BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

EXOPLANET DETECTION METHODS

METODY DETEKCE EXOPLANET

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR **NIKOL ŠKVAŘILOVÁ**
AUTOR PRÁCE

SUPERVISOR **Ing. TOMÁŠ KAŠPÁREK, Ph.D.**
VEDOUCÍ PRÁCE

BRNO 2025

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Zadání bakalářské práce

165024

Ústav: Ústav počítačové grafiky a multimédií (UPGM)
Studentka: **Škvařilová Nikol**
Program: Informační technologie
Název: **Metody detekce exoplanet**
Kategorie: Zpracování signálů
Akademický rok: 2024/25

Zadání:

1. Nastudujte metody detekce exoplanet. Porovnejte minimálně metodu tranzitní a metody využívající spektroskopická měření.
2. Zjistěte dostupné databáze naměřených dat pro výše zkoumané metody.
3. Navrhněte realizaci výše zkoumaných metod pro detekci exoplanet v rámci on-board zpracování na potenciální misi malé observatoře.
4. Porovnejte získané výsledky a diskutujte jejich budoucí možné rozšíření.

Literatura:

- Konacki, M., Torres, G., Jha, S. *et al.* An extrasolar planet that transits the disk of its parent star. *Nature* **421**, 507–509 (2003). https://doi.org/10.1038/nature01379
- M. Perryman, *The Exoplanet Handbook*, 2nd ed. Cambridge: Cambridge University Press, 2018. https://doi.org/10.1017/9781108304160
- David Charbonneau *et. al.* Detection of Planetary Transits Across a Sun-like Star, (1999), https://doi.org/10.1086/312457

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce: **Kašpárek Tomáš, Ing., Ph.D.**
Vedoucí ústavu: Černocký Jan, prof. Dr. Ing.
Datum zadání: 1.11.2024
Termín pro odevzdání: 14.5.2025
Datum schválení: 12.11.2024

Abstract

The aim of this work is to design and implement an approach for analysing data from the transit method for exoplanet detection. The data used comes from the TESS space telescope in the form of light curves, which record the observed brightness changes of a star over time. In this work, I focused on data-driven modelling and detection of exoplanets using Gaussian process regression. Both non-periodic and periodic kernel models were used to identify light curves suitable for detrending and to remove unwanted trends. Following this, models with periodic kernels were used to investigate the periodicity of the data. Finally, models with non-periodic kernels, trained on folded transits, were applied to detect transits in other light curves by correlation. The approaches described in this thesis provide an automated way to pre-process the data and identify possible transits.

Abstrakt

Cílem této práce je navrhnout a implementovat způsob analýzy dat z tranzitní metody pro detekci exoplanet. Použitá data pochází z vesmírného teleskopu TESS a jsou v podobě světelných křivek, které zachycují pozorované změny jasu hvězdy v čase. Tato práce se zaměřuje na modelování založené na datech pomocí regrese Gaussovských procesů. Modely s neperiodickými i periodickými jádry byly využity k identifikaci světelných křivek vhodných pro detrendování a následné odstranění nechtěných trendů. Dále byly použity modely s periodickými jádry ke zkoumání cyklických událostí v datech. Na závěr byly aplikovány modely s neperiodickými jádry, natrénovanými na složených tranzitech, k detekci tranzitů v ostatních světelných křivkách pomocí korelace. Postupy popsané v této práci nabízí automatizovaný způsob předzpracovní dat a detekci možných tranzitů.

Keywords

exoplanets, exoplanet detection methods, transit method, light curve, surrogate modelling, Gaussian processes, periodic and nonperiodic kernel, correlation

Klíčová slova

exoplanety, metody detekce exoplanet, tranzitní metoda, světelná křivka, náhradní modelování, Gaussovské procesy, periodický a neperiodický kernel, korelace

Reference

ŠKVAŘILOVÁ, Nikol. *Exoplanet detection methods.* Brno, 2025. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Ing. Tomáš Kašpárek, Ph.D.

Rozšířený abstrakt

Detekce a charakterizace exoplanet je čím dál aktivnějším oborem zaměřeným na objevování a poznávání nových světů. Tento obor se mimo jiné zabývá i hledáním života na planetách mimo naši sluneční soustavu. Se stávajícími a plánovanými misemi, které pozorují velké množství hvězd, není možné všechna data efektivně analyzovat ručně. Je proto zapotřebí automatizovaných metod, které z velkého kvanta dat vyberou ty hvězdy a systémy, které mají největší potenciál a jsou vhodné pro další prozkoumání.

Existuje řada fyzikálních metod pro detekci exoplanet. Většina spočívá v analýze určitého typu dat. Tranzitní metoda, která doposud odhalila největší množství exoplanet, sleduje měnící se pozorovaný jas hvězdy v čase. S přechodem exoplanety před hvězdou pozorovaný jas poklesne. Tato událost se nazývá tranzit. Data nabízená touto metodou jsou v podobě světelných křivek – seznamu záznamů obsahujících čas a naměřený jas. Vesmírné mise jako Kepler a TESS se zaměřují na detekci exoplanet pomocí této metody, a doposud vyprodukovaly světelné křivky pro více než milion hvězd.

Cílem této práce bylo navrhnout přístup, jak tato data automatizovaně analyzovat a hledat tranzity. Byla zvolena technologie náhradního modelování, konkrétně Gaussovských procesů v regresi. Postup se skládal z několika částí. První část se zabývala detekcí přítomnosti hvězdné proměnnosti, která do světelné křivky vnáší pomalý trend. Tento trend může značně narušovat schopnost následné detekce tranzitů, proto bylo nutné jeho přítomnost ve světelné křivce detekovat a poté odstranit, což bylo náplní druhého kroku. Poslední krok se věnoval samotné detekci tranzitů. Výsledný přístup využívá modelů tranzitů, které jsou vytvořeny ze složených křivek z hvězd se známými exoplanetami. V rámci detekce se vypočítá korelace mezi světelnou křivkou a modelem tranzitu. Výsledná korelace se poté porovná s vypočtenou hranicí. Na závěr práce byla provedena analýza více než sto tisíc světelných křivek. Výsledky této práce byly předány Astronomickému ústavu ČR, kde budou pro zajímavé kandidáty provedena další měření metodou radiálních rychlostí.

Jednou z největších překážek při návrhu postupu analýzy byla volba modelu pro zmíněný detrending, který zahrnuje odečtení natrénovaného modelu od původních dat. Důležitou součástí tohoto kroku byl výběr kernelu, který se pro model využije. Během trénování je snaha nalézt model, který nejlépe odpovídá předloženým datům. Toto však není během tvorby modelu pro detrending do jisté míry žádoucí, protože pokud model namodeloval kromě hvězdné proměnnosti i tranzity, byly následně odečteny od původních dat, a světelná křivka nebyla dále použitelná. Navrhovaný postup využívá přednastavování hyperparametrů kernelu, které kontrolují jeho citlivost na rychlé změny. Jsou porovnávány dva přístupy. První využívá pouze jednoho kernelu, a druhý využívá dvou kernelů, jeden pro modelování rychlých změn v datech a druhý pro modelování pomalých změn.

Výstupem práce je navržený postup pro kategorizaci a detrending světelných křivek a pro detekci tranzitů. Dále je výstupem knihovna v jazyce Python, umožňující flexibilní tvorbu vlastních zpracovávacích procesů. Navíc je k dispozici nástroj pro práci s touto knihovnou z příkazové řádky, který umožňuje snadné řetězení úloh a automatickou analýzu velkého množství dat a zobrazování výstupů této analýzy. V rámci práce jsou navíc dostupné pomocné skripty pro statistickou a vizuální analýzu výsledků produkovaných knihovnou.

Exoplanet detection methods

Declaration

I hereby declare that this Bachelor's thesis was prepared as an original work by the author under the supervision of Ing. Tomáš Kašpárek, Ph.D. The supplementary information was provided by RNDr. Petr Kabáth, Ph.D. I have listed all the literary sources, publications and other sources, which were used during the preparation of this thesis.

.....
Nikol Škvařilová
May 12, 2025

Acknowledgements

I would like to thank the thesis supervisor, Ing. Tomáš Kašpárek, Ph.D., for consultations and valuable advice. My appreciation also goes to RNDr. Petr Kabáth, Ph.D., for discussions and providing expert knowledge about exoplanets. Lastly, I would like to thank my brother for his support.

This thesis utilizes data collected by the Kepler mission and by the TESS mission, obtained from the MAST data archive at the Space Telescope Science Institute (STScI). Funding for the Kepler mission is provided by the NASA Science Mission Directorate, and for the TESS mission by the NASA Explorer Program. STScI is operated by the Association of Universities for Research in Astronomy, Inc., under NASA contract NAS 5–26555.

This thesis made use of Lightkurve, a Python package for Kepler and TESS data analysis (Lightkurve Collaboration, 2018) [23].

This thesis has made use of the NASA Exoplanet Archive, which is operated by the California Institute of Technology, under contract with the National Aeronautics and Space Administration under the Exoplanet Exploration Program.

Contents

List of Figures

List of abbreviations

BJD	Barycentric Julian Date. 18
CAL	Calibration. 16
CAS	Czech Academy of Sciences. 46
CCD	Charge-Coupled Device. 14
CFAR	Constant False Alarm Rate. 34
ESO	The European Organisation for Astronomical Research in the Southern Hemisphere. 12
ESPRESSO	Echelle Spectrograph for Rocky Exoplanet and Stable Spectroscopic Observations. 14
FFI	Full Frame Image. 15, 16
FFT	Fast Fourier Transform. 3, 34
GP	Gaussian Process. 19, 21
GPR	Gaussian Process Regression. 40, 43, 45, 46
IAU	International Astronomical Union. 8
JSON	JavaScript Object Notation. 37, 41, 44
MAST	Mikulski Archive for Space Telescopes. 15, 18, 41
PA	Photometric Analysis. 17
PDC	Presearch Data Conditioning. 17
PDCSAP	Presearch Data Conditioning Simple Aperture Photometry. 17, 18
PLATO	Planetary Transits and Oscillations of Stars. 14
SAP	Simple Aperture Photometry. 17, 18
SE	Squared Exponential. 3, 21–23, 25, 26, 28–30, 33, 37
SPOC	TESS Science Processing Operations Center. 3, 18, 36

STSCI Space Telescope Science Institute. 15

TCE Threshold Crossing Event. 17
TESS Transiting Exoplanet Survey Satellite. 3, 9, 14–18, 24, 36
TPF Target Pixel File. 15, 16

VLT Very Large Telescope. 14

Chapter 1

Introduction

The automatic detection of exoplanets is a relevant topic today, more than ever before, because of the significant amounts of data produced. Numerous space missions have been launched and are planned, not only with the goal of detection, but characterization as well. Increasingly more stars are being observed, therefore, it is necessary to employ automated data processing pipelines. Detected exoplanets can be further observed for characterization. This entails multiple disciplines, from understanding how solar systems form to the detection of life outside of Earth.

In this work, I learned about the most commonly used methods for exoplanet detection, with the main ones being the transit method and the radial velocity method. I chose to explore the transit method, which has detected the most candidates so far, and offers large quantities of data for analysis. The main data product from this method are light curves, which capture how the star's apparent brightness changes over time.

The text is structured as follows. Chapter 2 introduces the topic of exoplanet detection methods. It describes the principles behind the methods and outlines the transit method in detail, including relevant space missions and available data. Chapter 3 presents the basic principles behind Surrogate modeling, specifically Gaussian Process Regression, which is utilized in this work. Chapter 4 details the steps behind the analysis of the light curves, examined approaches, and results. The chapter also sets out the final pipeline, used to analyze over a hundred thousand light curves. Chapter 5 explains the implementation of the solution, highlighting the most important concepts. Finally, Chapter 6 discusses the potential improvements and extensions of the current solution.

Chapter 2

Exoplanets

In this chapter, the most important concepts from the field of exoplanet detection are introduced. The description of the commonly used methods is followed by details about two space missions related to the transit method, along with their main data products, as these are utilized in this work. Some of the workings of the instruments are illustrated as well, including the standard data processing pipeline.

2.1 Definition of an exoplanet

An exoplanet is a term used for planets outside our Solar system. The frequently quoted working definition from the International Astronomical Union (IAU) puts up the following mass-related bounds [34, p. 5]:

1. The minimal mass is the same as for planets in our Solar System – thereby it must be enough for the object to be nearly round and clear its surroundings.

2. The maximal mass is 13 Jupiter masses.

2.2 History of exoplanet discovery

NASA offers a comprehensive walk-through of the history of exoplanet detection on their website [27]. The initial discovery was made in 1992 when two exoplanets were identified in orbit around a pulsar. Subsequently, in 1995, an exoplanet was discovered in orbit around a Sun-like star 51 Pegasi. In 1999, the first transit was detected.

2.3 Methods for exoplanet detection

In total, 5893 exoplanets have been detected and confirmed [28] so far. Table 2.1 shows the number of exoplanets detected by various methods. Figure 2.1 shows the cumulative count of exoplanets detected by different methods over the years, highlighting the importance of the transit and radial velocity methods.

Discovery method	Number of Exoplanets
Transit	4374
Radial Velocity	1119
Microlensing	239
Imaging	83
Transit timing variations	36
Eclipse timing variations	17
Orbital brightness modulations	9
Pulsar timing variations	8
Astrometry	5
Pulsation timing variations	2
Disk Kinematics	1

Table 2.1: Number of exoplanets detected by various methods. Source: [28]

.

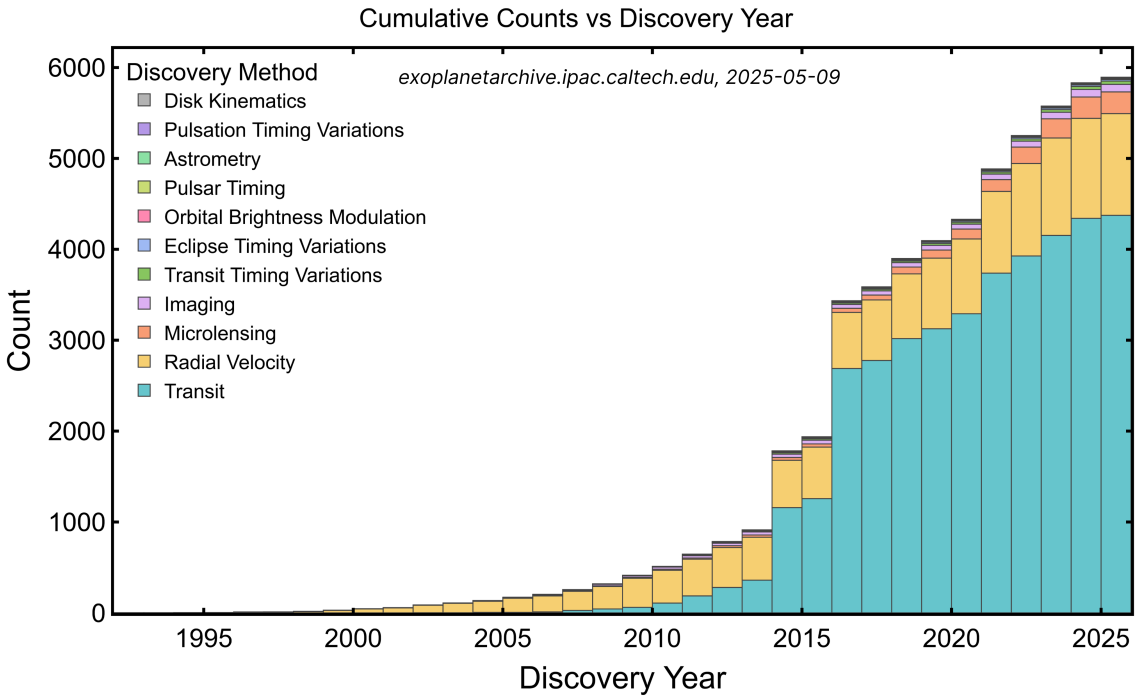

Figure 2.1: Cumulative count of detected exoplanets by various methods over the years. Source: [29].

2.3.1 Transit method

The transit method is the most used method for exoplanet detection, and has currently identified more than 4000 exoplanets.

Space missions, such as the Transiting Exoplanet Survey Satellite (TESS), observe a set of stars for a period of time. When an exoplanet passes in front of the star, it blocks some portion of the light from the star, and the star appears dimmer, as is shown in Figure 2.2. In the resulting light curve (star's flux evolution in time), the periodic dips mark the presence of another object [6, p. 89].

For this method to effectively detect an exoplanet, the exoplanet must orbit so that it passes in front of the star from the observer's point of view. If this is not the case, no light from the star is blocked, and the exoplanet does not produce any dips in the light curve, which would give away its presence [6, p. 89].

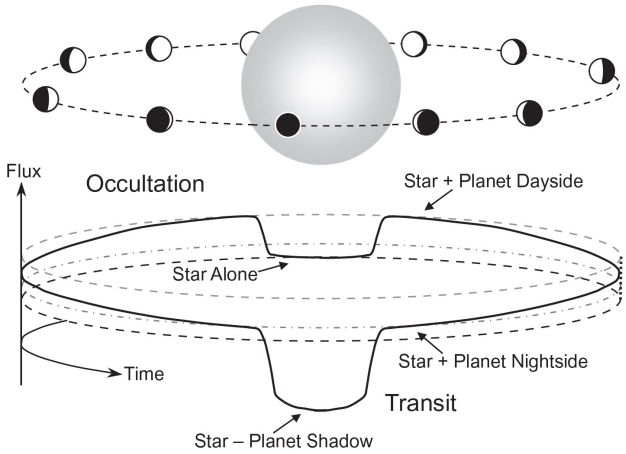

Figure 2.2: The impact of a transiting exoplanet on the star's light curve, producing a dip in observed brightness. Source: [44, p. 56].

2.3.2 Radial velocity method

The radial velocity method, or spectroscopic method, is significant historically and even today. It is the second method in the discovery of exoplanets by number of detections, and it was the first method to detect an exoplanet around a Sun-like star. With improving instruments, this method is able to detect smaller changes in radial velocities than ever before, making it possible to detect even smaller exoplanets.

To detect an exoplanet using this method, the light (spectrum) of the star is observed over a period of time. When an exoplanet orbits a star, its gravitational pull misplaces the star. This displacement of the star, as it orbits the shared center of gravity, is visible in the spectrum due to the Doppler effect. When the star moves toward the observer, the spectrum gets blue-shifted, and when it moves away from the observer, the spectrum gets red-shifted. This shift signals the presence of a different body influencing the star – potentially an exoplanet [17, p. 4]. Figure 2.3 shows the radial velocity of a star gained as a result of an orbiting exoplanet.

Based on the magnitude, periodicity, and other parameters of the shift in the spectrum, the minimal mass of the exoplanet can be estimated, along with the eccentricity of its orbit. To get a better estimate of its mass, data from other methods are typically needed. Usual targets for this method are heavy exoplanets orbiting in close proximity to their host star, as pointed out in [19, p. 2].

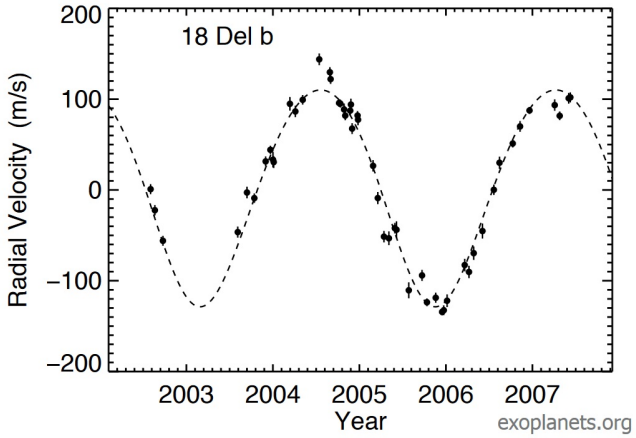

Figure 2.3: Radial velocity measurements of a star (18 Delphini), which is orbited by an exoplanet. Source: [9].

2.3.3 Astrometry

Astrometry utilizes the gravitational pull from an exoplanet orbiting a star. The star then moves in a different fashion than it would otherwise if there was no exoplanet. This difference between the original and the predicted trajectory indicates the presence of possibly an exoplanet [30, p. 81].

Figure 2.4 shows this discrepancy on a model, where the solid line differs from the predicted dotted line.

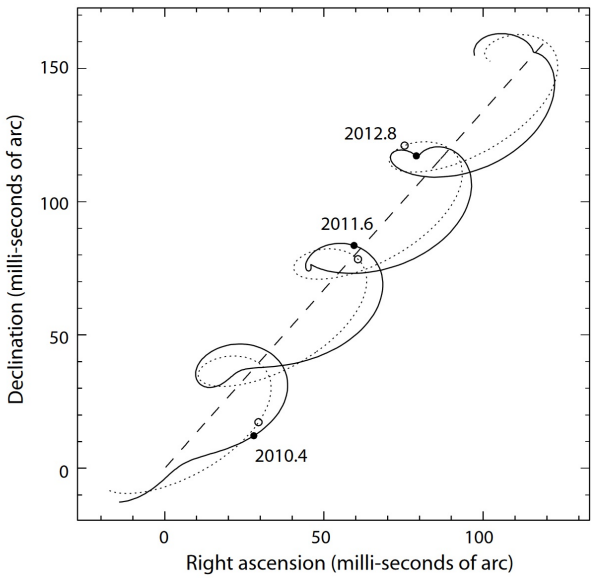

Figure 2.4: Gravitational pull from the exoplanet applied on the star (solid line) as opposed to the expected behavior (dotted line). Source: [31].

2.3.4 Direct imaging

Exoplanets are usually not visible on their own, as they are dim compared to the host star. To capture them directly, the exoplanet must reflect or produce light [30, p. 329]. Intuitively, as pointed out in [30, p. 330], the typical targets include large exoplanets, which are capable of reflecting enough light so that it is distinguishable from the star or exoplanets orbiting young stars and early in their development, which are self-luminous. To capture an image such as 2.5, the exoplanet also has to orbit at a significant distance from the star.

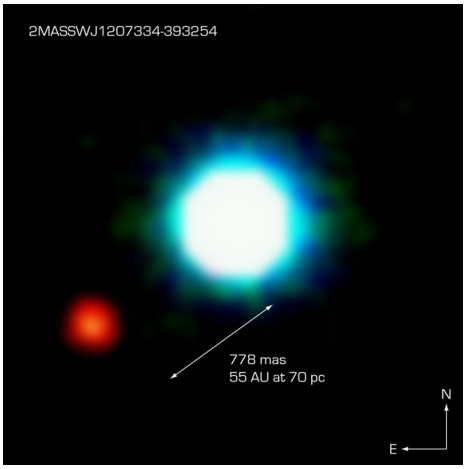

Figure 2.5: First image of an exoplanet (2M1207 b). Source: ESO [8].

2.3.5 Microlensing

Microlensing is generally a rare event [14, p. 135], where a star, or another light source, passes behind the target star. During this event, the star in the front gravitationally bends the light from the background object, and the observed brightness is increased. If an exoplanet orbits the star in the front, it too bends the light from the background, and as a result, sharp peaks are visible in the brightness of the event [19, p. 6].

This change is shown in Figure 2.6, where the exoplanet caused the visible sharp peaks. By their nature, these events usually occur only once, therefore, multiple observations are not possible. Still, over 200 exoplanets have been detected this way.

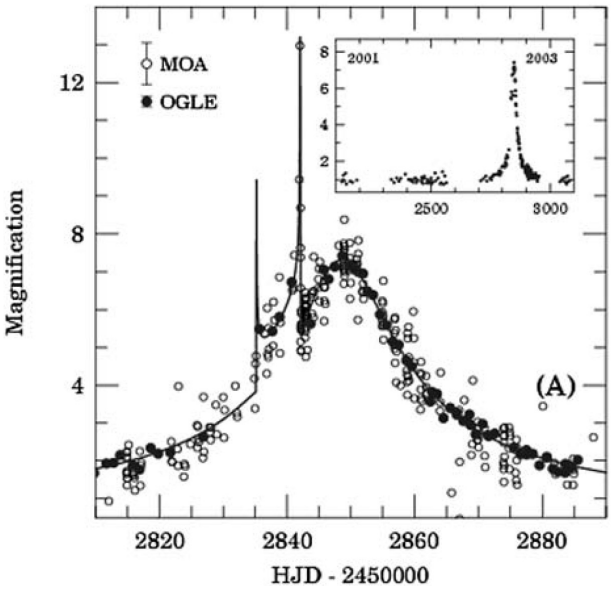

Figure 2.6: Exoplanet orbiting a star produces sharp peaks during a microlensing event. Source: [19, p. 7].

2.3.6 Timing variations

The first exoplanet was detected using the timing variations method around a pulsar. Some objects in space produce a highly periodic signal, e.g., pulsars, pulsating stars, or eclipsing binaries. An exoplanet orbiting such a system or an object will change the location of the shared gravitational center, which then induces changes in the arrival time of said signals [30, p. 103].

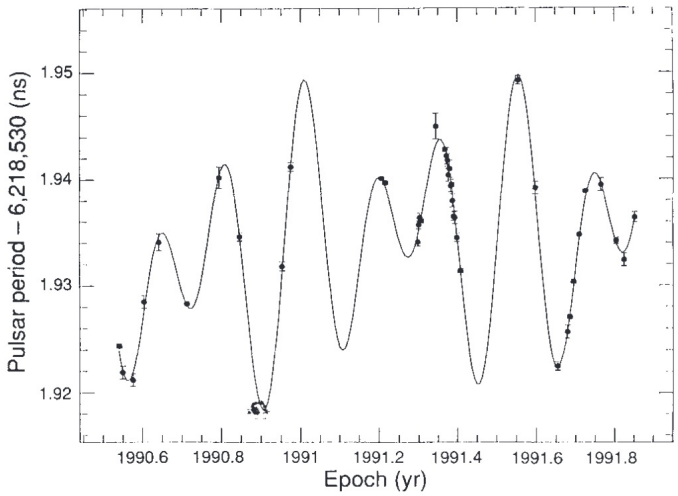

Figure 2.7: Variations in the time between pulses of PSR1257+12. Source: [45].

2.4 Missions and telescopes

Kepler/K2 and TESS space missions are the most significant for exoplanet discovery via the transit method. Kepler/K2 operated from 2009 to 2018; TESS launched in 2018 and is still continuing its observations. Another mission, planned to launch in 2026, is the Planetary Transits and Oscillations of Stars (PLATO)[1].

2.4.1 Kepler

The Kepler mission is described in the Kepler Instrument Handbook [40]. Throughout its mission, Kepler surveyed more than 100,000 stars with the goal of detecting transits of Earth-like exoplanets orbiting at a sufficient distance from their host star, so that they could potentially host life (the habitable zone). The telescope was equipped with 21 Science CCD sensors, pixels from which were coadded in the Science Data Accumulators. Due to Kepler's limited memory, mostly pixels containing selected targets and calibration pixels were stored onboard to be downlinked to Earth. Every quarter, the telescope performed a 90° roll to keep the solar panels pointed at the Sun and the radiator pointed into deep space – data batches from Kepler are divided by quarters [40, p. 15].

2.4.2 K2

K2 started in 2014 and was an extension of Kepler's primary mission. It is described in detail in the K2 Handbook [26]. During this extension, Kepler observed different portions of the sky around the ecliptic, as it could no longer observe the original portion of the sky due to the failure of the second of the four reaction wheels [26, p. 7-8].

2.4.3 TESS

The TESS mission is described in the TESS Instrument Handbook [42]. The telescope is equipped with four wide-field CCD cameras. In a similar fashion to Kepler, data from these cameras is coadded and then stored for downlink. TESS offers data in the form of pixels around target stars and images encompassing the whole field of view, which are produced continuously [42, p. 11].

2.4.4 Ground telescopes

While space telescopes provide a significant advancement in the field of exoplanet detection, ground-based observatories still play an important role. The Very Large Telescope (VLT) in Chile, with instruments such as the Echelle Spectrograph for Rocky Exoplanet and Stable Spectroscopic Observations (ESPRESSO) [7], is capable of precise radial velocity measurements, enabling the detection of Earth-like exoplanets. The Perek's telescope, situated in Ondřejov in the Czech Republic, is also used for exoplanet confirmation and characterization [21].

[1] https://www.esa.int/Science_Exploration/Space_Science/Plato

2.5 Available data from Kepler/K2 and TESS

Time series data from both Kepler/K2 and TESS are available at the Mikulski Archive for Space Telescopes (MAST) at the STSCI[2]. Detected planetary candidates are available at the NASA Exoplanet Archive at the NASA Exoplanet Science Institute[3].

The main data products from both telescopes are the following:

1. Full Frame Images (FFIs), which contain the entire field of view of the telescope.
2. Target Pixel Files (TPFs), which contain raw pixel data (postage stamps) for individual target stars.
3. Light Curve Files, which are created from the TPFs and capture the star's flux in time.

2.5.1 Kepler/K2

The Kepler Data Characteristics Handbook [41] provides a description of the available data. Cadence corresponds to a time-stamped co-added image from multiple exposures. The cadence period defines how many seconds of exposure were co-added to create the final image (FFIs, TPFs). The data from Kepler comes in two cadence periods: the long cadence, corresponding to 270 frames (29.425 min), and the short cadence, corresponding to 9 frames (58.85 s). Frames had an exposure time of 6.02s and 0.52s read-out time [41, p. 10].

During the primary mission, Kepler monitored a small part of the sky near the constellation of Cygni, shown in 2.8.

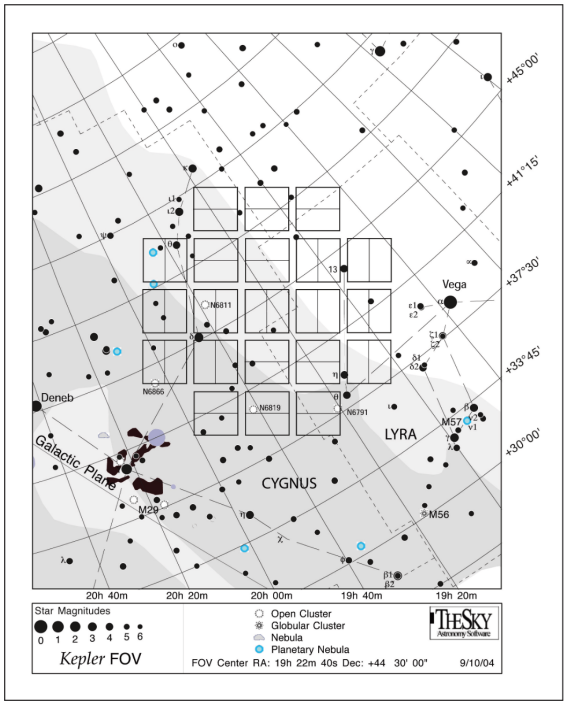

Figure 2.8: Kepler Field of View. Source: [40, p. 17]

[2] https://archive.stsci.edu/
[3] https://exoplanetarchive.ipac.caltech.edu/

K2 observed different portions of the sky in about 83 days long campaigns [26, p. 8], shown in Figure 2.9.

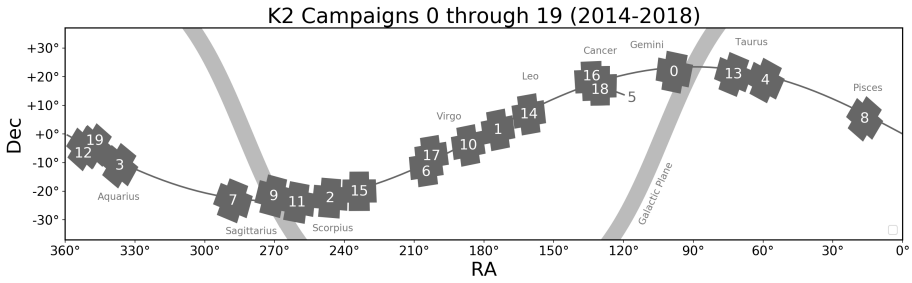

Figure 2.9: K2 campaigns. The thin line represents the ecliptic plane. Source: [26, p. 8]

2.5.2 TESS

TESS data products are described in the TESS Handbook [42]. TESS observes stars in sectors – $24° \times 96°$ regions of sky. Each sector is monitored for about 27 days [42, p. 14]. After completing the primary mission at the end of the first two years, sectors 1 to 26 were observed, encompassing the whole sky, as seen in Figure 2.10. TESS then continued to observe other sectors of the sky, which overlap with the sectors from the first two years, but are marked with a higher number. This means that a star can be observed in more sectors with different numbers. On top of that, adjacent sectors overlap more significantly around the poles, which also results in more observation time for various stars.

As described in [42, p. 16], TESS produces TPFs with a cadence period of 60 frames (2 mins) from 2-second exposures. FFIs were produced from 900 frames (30 min).

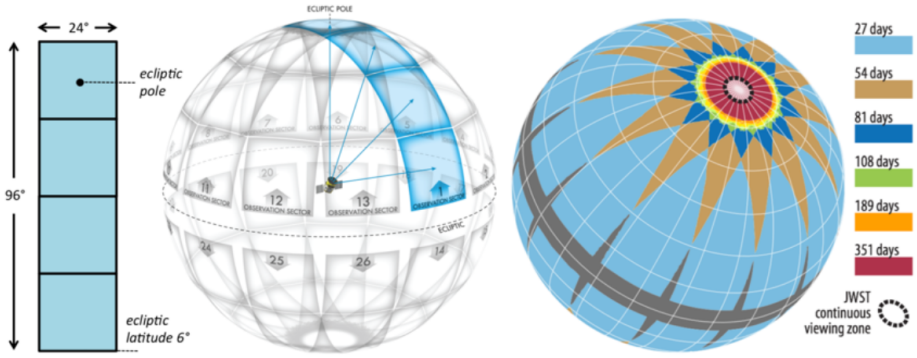

Figure 2.10: Due to the sector overlap, stars closer to the poles have been observed for longer periods than the standard 27 days. Source: [42, p. 15].

2.6 The Kepler Pipeline

The Kepler pipeline describes how raw data from the spacecraft is processed. The main science modules, visualized in Figure 2.11, are the following [20]:

1. Calibration (CAL) – corrects raw photometric data from the spacecraft.

2. Photometric Analysis (PA) – extracts Simple Aperture Photometry (SAP), which includes removing the light effect of the background from the target pixels and summing them. This module produces SAP light curves.

3. Presearch Data Conditioning (PDC) – removes systematic errors from SAP light curves caused by pointing errors, focus changes, thermal effects, etc. This module produces PDCSAP light curves.

4. Transiting Planet Search (TPS) – identifies Threshold Crossing Events (TCEs), which are parts of the PDCSAP light curve showing notable characteristics of an event (exoplanet transit, eclipse, etc.).

5. Data Validation (DV) – fits transit models to TCEs, producing various statistics and reports. After a TCE is processed, it is removed from the light curve so that other TCEs can be identified and processed. As a result of fitting a transit model to the TCE, the physical attributes of the exoplanet can be estimated. TCEs are then subjected to vetting.

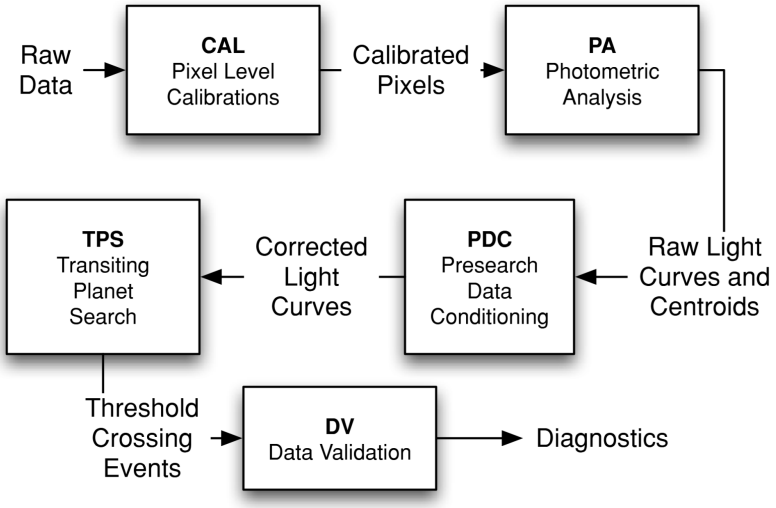

Figure 2.11: The Kepler pipeline. Source: [20]

The first three modules create archived data products, mentioned in 2.5. The other two modules are concerned with exoplanet detection and producing reports and TCEs, which are archived for further analysis and made available to the public as well.

2.7 Working with light curves

To download light curves, I used the Python library `lightkurve`. The usage is discussed in Section 5.5. Figure 2.12 shows light curves for different stars captured by TESS.

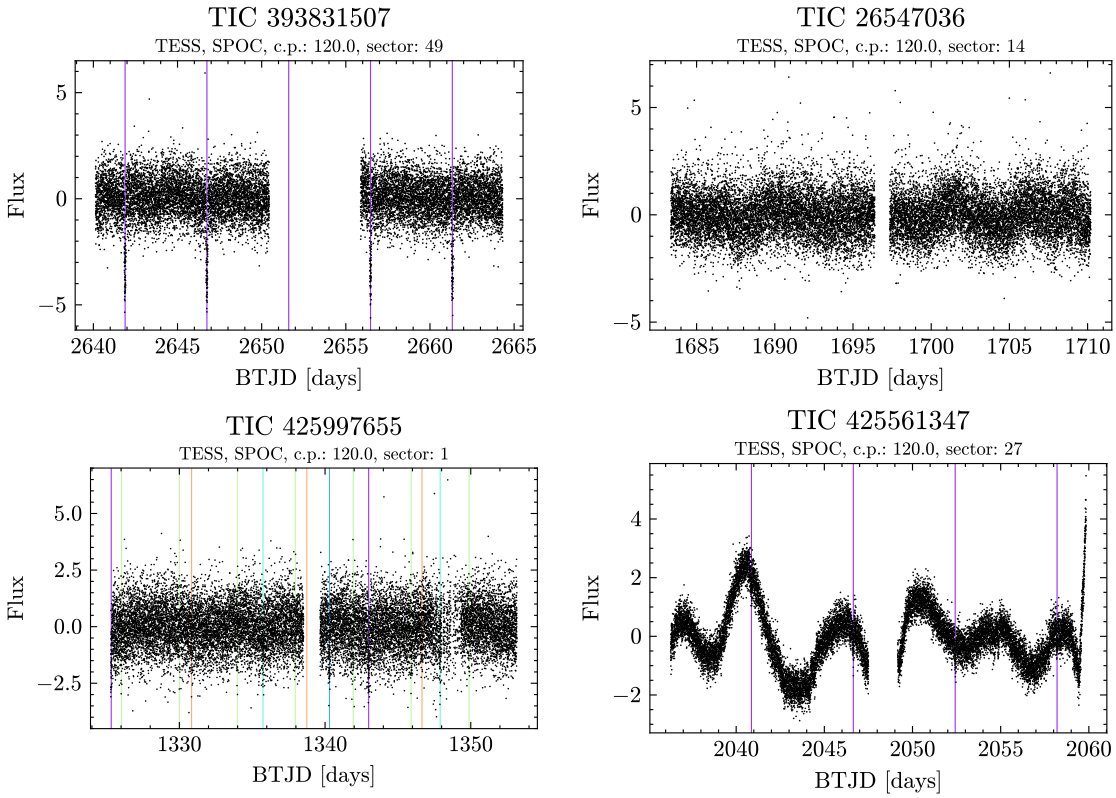

Figure 2.12: Examples of light curves for different stars from TESS. Vertical lines mark transit positions, color differentiates between exoplanets. Some light curves contain clear transits, while in other cases, the transits are much more subtle. At the same time, multiple exoplanets may transit during a single sector. Stellar variability can be visible in the light curves, presenting as a slow trend.

Light curves with relevant metadata are typically stored in a FITS file format as data points. Each data point contains a timestamp, the PDCSAP flux, the SAP flux, a quality flag, and other information described in the MAST Kepler Archive Manual [38]. For TESS, the data format is described in the Science Data Products Description Document[4]. The timestamp for both Kepler and TESS is in the form of a Barycentric Julian Date (BJD), offset by some specified value [41, p. 46] [42, p. 61].

As shown in Figure 2.12, light curves may contain gaps, which can be caused by various reasons. Examples include data downlink to Earth, during which the instrument did not produce any data, or if the data points are corrupted – they are available, but have non-zero quality flag and are automatically ignored by `Lightkurve`[5].

This work analyzed PDCSAP light curves with a two-minute cadence period from TESS, which were produced by SPOC [5]. The data may be obtained from doi:10.17909/t9-wpz1-8s54.

[4] https://archive.stsci.edu/files/live/sites/mast/files/home/missions-and-data/active-missions/tess/_documents/EXP-TESS-ARC-ICD-TM-0014-Rev-F.pdf

[5] https://lightkurve.github.io/lightkurve/tutorials/2-creating-light-curves/2-2-kepler-noise-1-data-gaps-and-quality-flags.html

Chapter 3

Surrogate modeling

The goal of surrogate modeling, as described in [11], [22], is to create a model that approximates the behavior of an exact model. Exact models can be difficult to produce, especially when modeling complex systems or systems with unknown principles that govern them.

Surrogate modeling is often used in engineering, where simulations using exact models might be unfeasible due to time constraints. Surrogate models (surrogates) are also applicable in situations where only a small number of observations is available, either due to the nature of the observed object or due to technical/financial limitations [11].

Surrogate modeling is a supervised learning approach. This means that the model learns from labeled data. For regression problems, the training data include the input values (e.g., timestamps) and expected output values (e.g., flux). Commonly used algorithms [10] include polynomial models, radial basis function models, kriging (Gaussian process regression), and support vector regression.

3.1 Gaussian processes

A Gaussian Process (GP), as described in [33, chapter 1.1], is based on Bayes' theorem. GP assumes a multivariate Gaussian distribution, which is fully specified in prior by the mean function and covariance function (kernel). The mean function and kernel enforce prior assumptions about the data, such as linear or periodic trends. This means that just with the selection of the mean function, kernel, and priors, it is possible to encode some expected behavior directly into the model, without any data. Examples of this might include constraining the kernel hyperparameters to a certain range of values, coming from information about the system. By introducing data and applying Bayes' theorem, the prior is updated, and the posterior is obtained. Functions sampled from the posterior match the observed data as well as the priors set out by the kernel and mean function. The benefit of a GP is that a prediction is a probability distribution with mean and variance [33, chapter 1]. This means that it is possible to study the model's confidence in its various sections. Figure 3.1 shows an example of functions sampled from the prior and the posterior, along with the final model and 95% confidence interval.

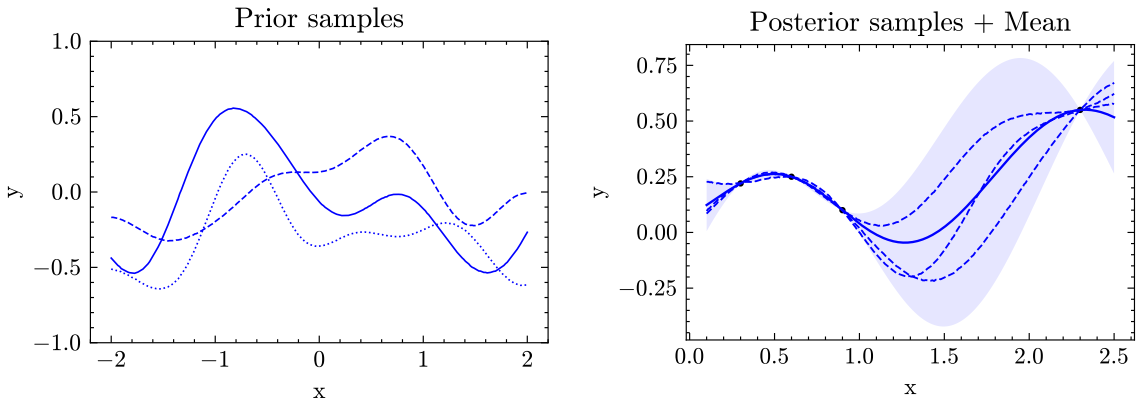

Figure 3.1: Sampled functions from the prior and the posterior, and the final model.

The following definitions and equations are taken from the book Gaussian Processes for Machine Learning [33, chapter 2.2].

The mean function and kernel are defined as

$$m(\mathbf{x}) = E(f(\mathbf{x})), \tag{3.1}$$

$$k(\mathbf{x}, \mathbf{x}') = E((f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))), \tag{3.2}$$

where $\mathbf{x}$, $\mathbf{x}'$ represent input points.

GP is fully defined by its mean function and kernel; therefore

$$f(\mathbf{x}) \sim GP(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')). \tag{3.3}$$

The mean function is often set to a constant function at 0:

$$m(\mathbf{x}) = \mathbf{0}. \tag{3.4}$$

To sample functions from the prior, input points X_* are selected, at which the function is sampled. Using the kernel, the covariance matrix is then created as $K(X_*, X_*)$. The function values $\mathbf{f}_*$ at X_* are sampled from the multivariate Gaussian distribution as follows:

$$\mathbf{f}_* \sim \mathcal{N}(\mathbf{0}, K(X_*, X_*)). \tag{3.5}$$

This step involves performing Cholesky decomposition on the covariance matrix, generating independent samples from a standard normal distribution, and multiplying these samples by the Cholesky matrix. This produces the correlated samples.

To sample data from the posterior, the joint distribution of training and test outputs is created as follows:

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X, X) & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right) \tag{3.6}$$

After observing the data, conditioning is required to construct the posterior distribution, leveraging Bayes' theorem. Conditioning limits the generated functions so that they correspond to the observed data.

$$\mathbf{f}_* | X_*, X, \mathbf{f} \sim \mathcal{N}(\mu_*, \Sigma_*) \tag{3.7}$$

where

- $\mu_* = K(X_*, X)K(X, X)^{-1}\mathbf{f}$,
- $\Sigma_* = K(X_*, X_*) - K(X_*, X)K(X, X)^{-1}K(X, X_*)$.

Sampling functions from the posterior is performed in the same way as sampling from the prior.

Typically, the original function is not known. Available samples include noise, denoted as ϵ in

$$y = f(\mathbf{x}) + \epsilon. \tag{3.8}$$

It is assumed that the noise has a Gaussian distribution, with variance σ_n^2. The joined prior distribution can then be written as

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X, X) + \sigma_n^2\mathbf{I} & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right). \tag{3.9}$$

The posterior distribution then corresponds to

$$\mathbf{f}_* | X_*, X, \mathbf{f} \sim \mathcal{N}(\mu_*, \Sigma_*) \tag{3.10}$$

where

- $\mu_* = K(X_*, X)[K(X, X) + \sigma_n^2\mathbf{I}]^{-1}\mathbf{y}$,
- $\Sigma_* = K(X_*, X_*) - K(X_*, X)[K(X, X) + \sigma_n^2\mathbf{I}]^{-1}K(X, X_*)$.

The final mean represents the best guess, acknowledging the prior assumptions and observed data. The final covariance specifies the uncertainty, which decreases near data points and increases in regions without data points.

GPs are generally computationally expensive due to the inversion step during conditioning, making it $\mathcal{O}(n^3)$.

3.1.1 Kernel and mean function

A covariance function, also called a kernel, is a function that calculates the similarity between two inputs [33, p. 79].

Kernels can be tuned by hyperparameters, such as length scale and variance. Each kernel has its own set of hyperparameters. The process of training a GP involves selecting the kernel and a combination of hyperparameters for the kernel that fits the observed data best [33, p. 106].

Commonly used kernels include the periodic kernel, polynomial kernel, and the Matérn family (including the Squared Exponential (SE) kernel). Figure 3.2 shows examples of different kernels as functions sampled from the prior.

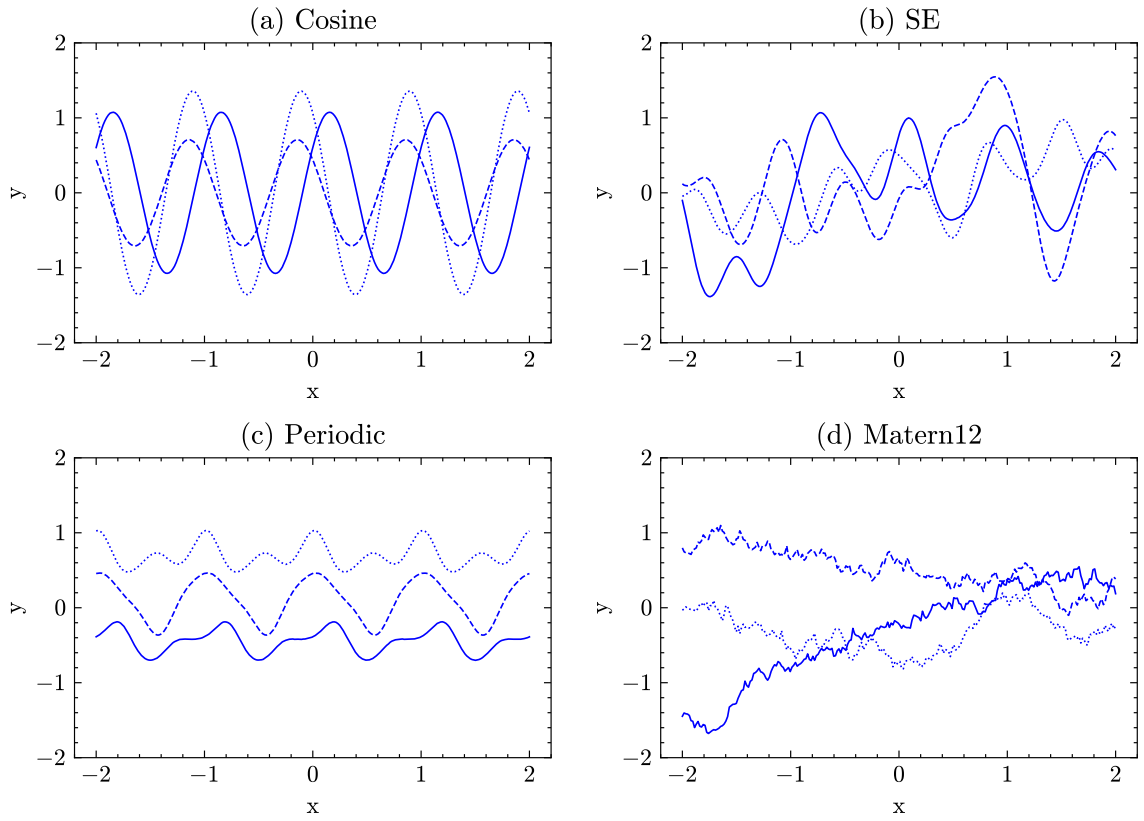

Figure 3.2: Samples from the prior created using (a) the cosine kernel, (b) SE kernel, (c) the periodic kernel, (d) the Matern12 kernel.

Kernels can be combined by addition and multiplication [33, p. 95]. Figure 3.3 shows functions sampled from the prior for different combinations of commonly used kernels.

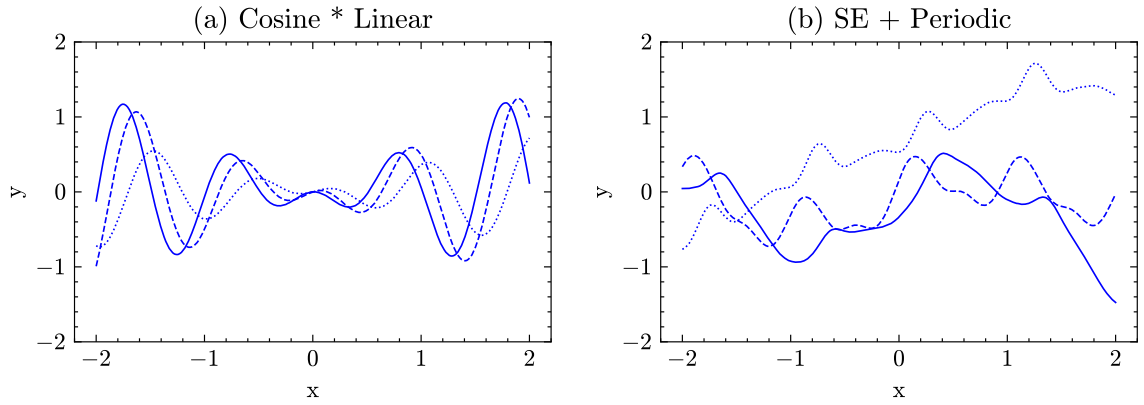

Figure 3.3: Prior samples created using the (a) multiplication of cosine and linear kernel, (b) addition of SE kernel and periodic kernel.

Kernel creates the covariance matrix from inputs. Inputs considered similar by the kernel produce large positive numbers, while dissimilar inputs produce small negative numbers. Figure 3.4 shows covariance matrices for different kernels.

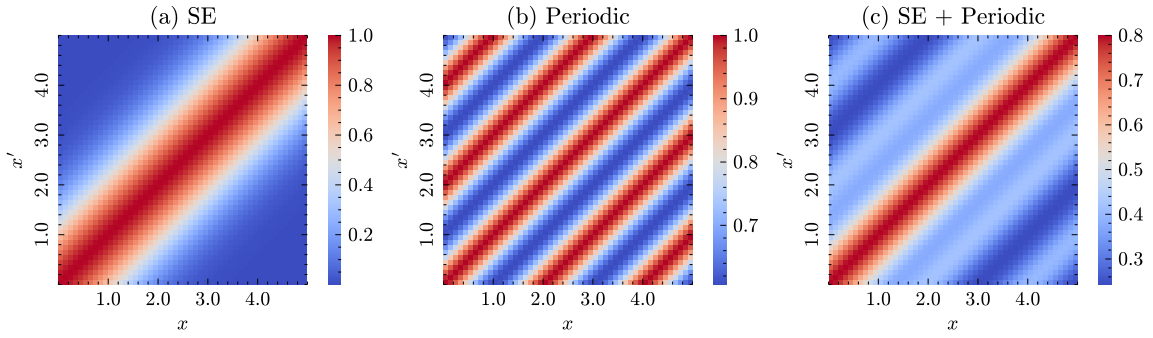

Figure 3.4: Covariance matrices for (a) SE kernel, (b) periodic kernel, and (c) addition of SE kernel and periodic kernel.

The mean function also encodes prior assumptions about the data into the model. The prediction is centered around the mean function, as is visible in Figure 3.5.

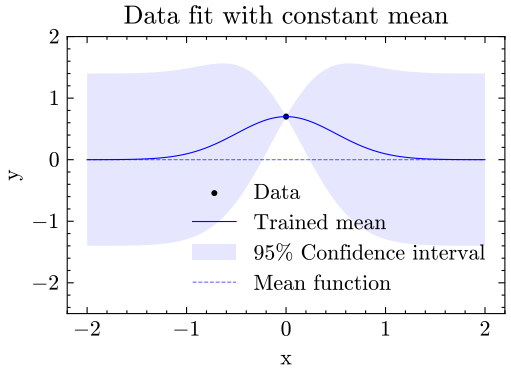

Figure 3.5: Data fit constructed from a single data point, using SE kernel and constant mean function.

Hyperparameters are optimized during the training to obtain the best-fitting model. The SE kernel, has two hyperparameters: variance (v) and length scale (l). Variance controls how far the model deviates from the mean. Length scale indicates the smoothness of the model. Figure 3.6 shows examples of the SE kernel with different length scales.

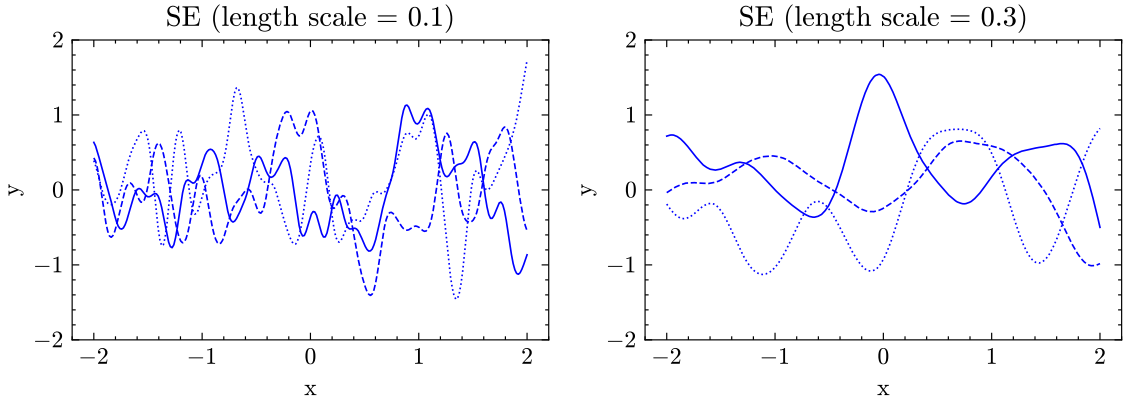

Figure 3.6: Two examples of the SE kernel with different length scales.

Chapter 4

Analysis

In this section, the steps of the analysis are described, including the various approaches for each one. The first step was concerned with identifying which light curves contain stellar variability and therefore need detrending. The following step was the detrending itself, and the last step included the detection of transits.

After the initial analysis, I constructed a data processing pipeline with selected approaches from each step. Over a hundred thousand light curves were evaluated, the results for which are discussed at the end of this chapter.

For this analysis, the stars were chosen from the TESS Project Candidates table [15], which I downloaded through the NASA Exoplanet Archive[1]. In total, 405 stars with at least one confirmed exoplanet were selected.

4.1 Stellar variability

Transits are not the only event changing the apparent brightness of a star. Stellar variability contributes to the light curve as a slow trend and can be caused by various processes, such as sunspots [39]. In this section, I analyze the stellar variability in light curves for categorization and detrending.

4.1.1 Defined light curve categories

To be able to examine the data in the context of stellar variability, I labeled the data by hand using `LabelMe` [43] Software. Each light curve has at least one label from the following list:

- Major Activity – The light curve shows significant stellar variability and is a potential candidate for detrending.
- Minor Activity – The light curve shows minor stellar variability and could also benefit from detrending.
- Clear Transits – The light curve contains clear transits, distinguishable by eye.
- Noisy – The light curve is noisy; it is difficult to detect transits by eye.

[1] https://exoplanetarchive.ipac.caltech.edu/cgi-bin/TblView/nph-tblView?app=ExoTbls&config=TOI

A single light curve could get multiple labels; because of this, in the following plots, labels such as „Clear Transits“ are omitted if the light curve is also labeled with any stellar variability. Figure 4.1 shows examples of light curves for each label.

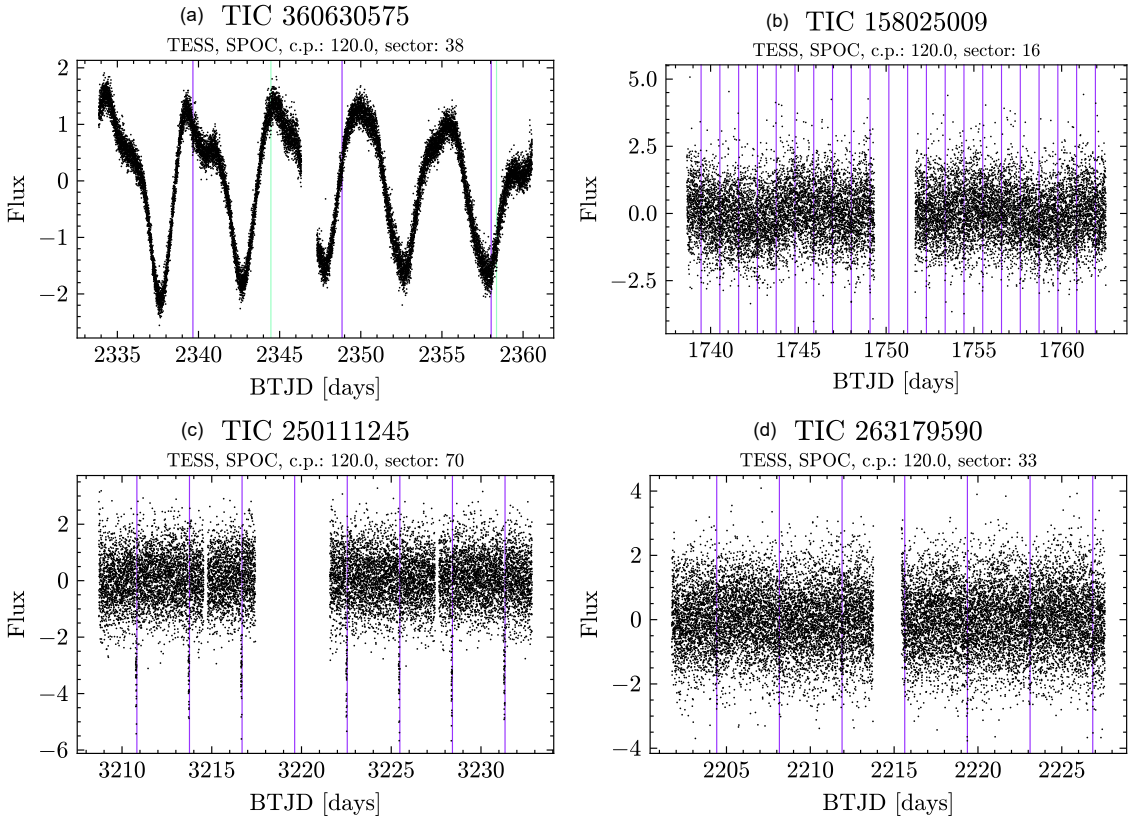

Figure 4.1: Light curves for different stars labeled (a) Significant Activity, (b) Minor Activity, (c) Clear Transits, and (d) Noisy.

4.1.2 Light curve classification process

To identify light curves suitable for detrending, I examined two approaches. In the first approach, models with different kernels were trained, attributes from which were then examined, especially the logarithmic marginal likelihood and its variance.

Most notably, the SE kernel and the SE + Periodic kernel resulted in higher likelihood values and lower variance for light curves with significant activity, as is shown in Figure 4.2. At the same time, models with noisy data typically got lower likelihood values and higher variance.

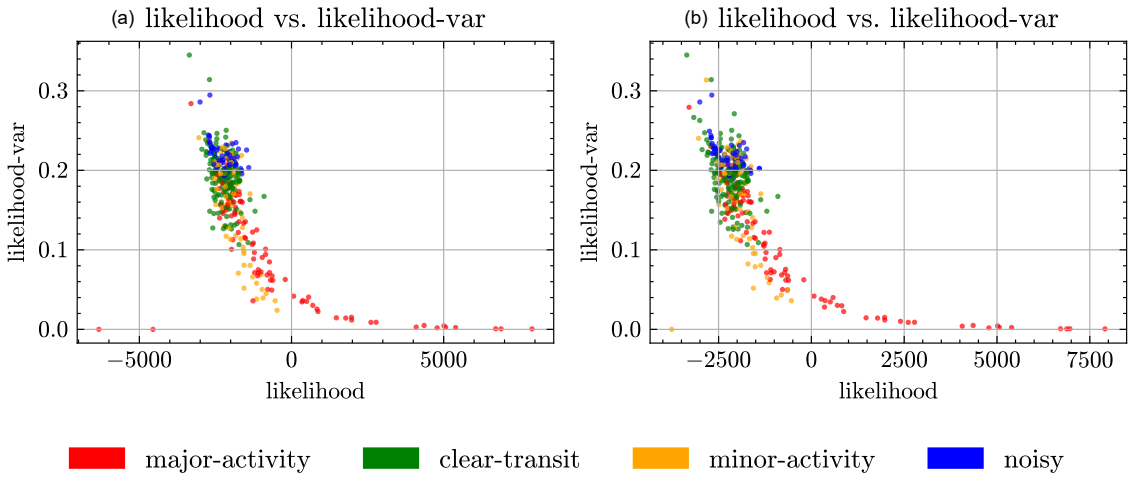

Figure 4.2: Logarithmic marginal likelihood and its variance from models with the (a) SE kernel and (b) SE + periodic kernel. Significant outliers were removed from the figures for better visualization.

The pipeline for this approach is visualized in Figure 4.3. The categorization process starts with a light curve. A model with a chosen kernel is then created and trained. From this model, the log. marginal likelihood is used to separate light curves suitable for detrending from the data without trends.

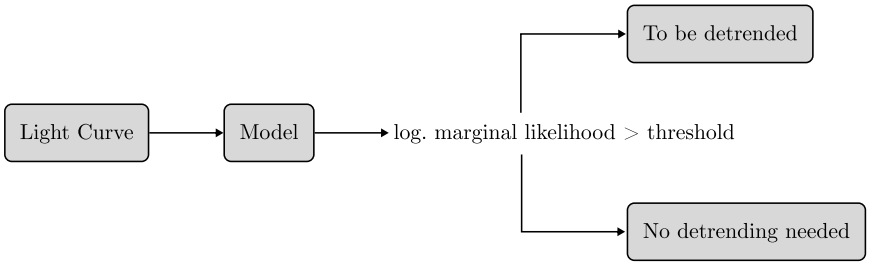

Figure 4.3: The categorization pipeline using log. marginal likelihood.

Another approach, directly tied to detrending, utilizes the fact that already flat light curves will not be as different from their detrended counterparts. This difference between the original light curve and the detrended version, if significant, can indicate that the original light curve contained considerable activity, which the detrending model captured and removed. The results of this approach are summarized in this section, while the topic of choosing the proper detrending model is discussed in the following section. Figure 4.4 shows examples of two light curves – their original flux values, detrended values, and their differences.

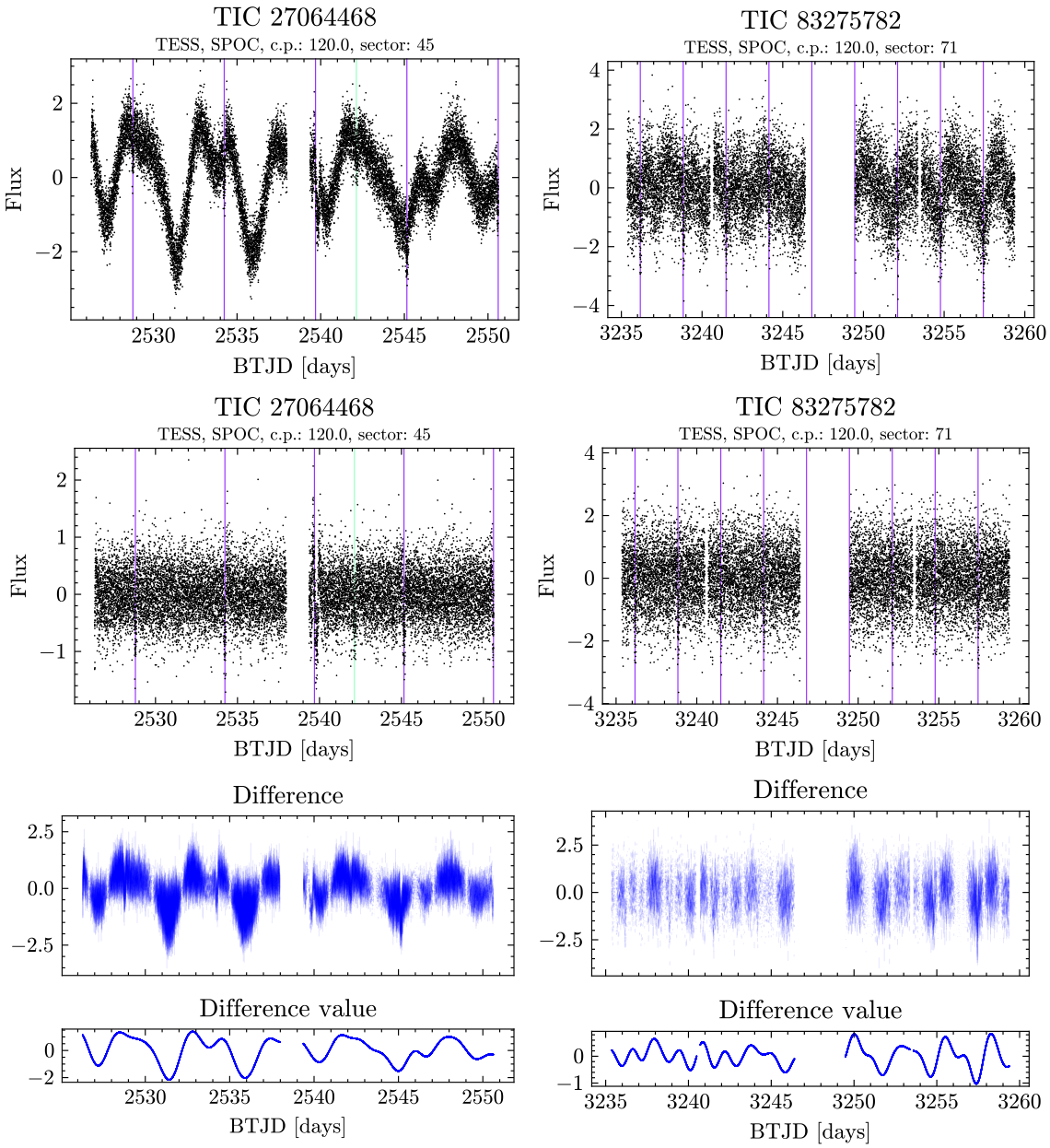

Figure 4.4: Examples of two stars, showing the original light curve in the first row, the detrended version in the second, and the difference in the last two.

The mean calculated from the absolute difference between the light curves is then used to indicate whether the light curve needs detrending, as light curves with significant stellar variability are detrended more heavily than flat light curves. Therefore, the mean absolute difference is higher. Figure 4.5 shows results for this approach. Detrended flux subtracted from the original flux, taken as the mean of absolute values over the light curve.

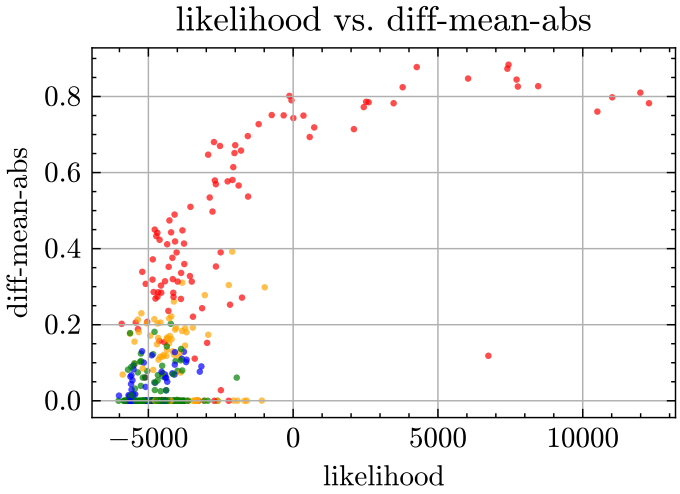

Figure 4.5: Mean absolute difference between detrended and original light curve.

4.1.3 Data detrending

After a model is trained, it can produce predictions. This prediction can be made over the time stamps of the original light curve. The predicted flux values are then subtracted from the actual flux values, producing a new light curve. Figure 4.6 illustrates the detrending process. The detrending pipeline utilizes two variations of the same light curve, downsampled (which helps with the smoothing of the model) and the original version. Prediction from a trained model is subtracted from the original light curve, producing new detrended light curve.

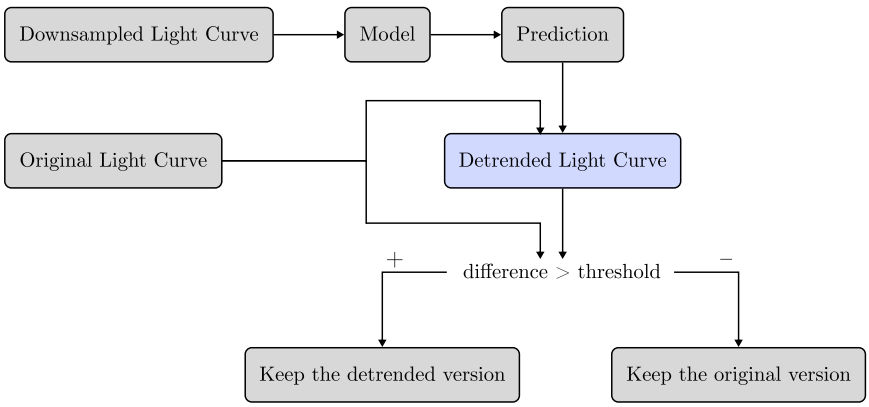

Figure 4.6: The detrending pipeline.

Mostly, the SE kernel was used in the experiments. When the model was trained without constraints, the prediction closely matched the original light curve. Subtraction of this prediction either removed the transits from the light curve or heavily distorted them. This led to a light curve not usable for further analysis.

To make the model smoother, constraints had to be put on the kernel's hyperparameters before the training. The SE kernel comes with two hyperparameters, the variance and length scale. The length scale controls the smoothness of the model; the larger the length scale, the smoother the model, as discussed in Section 3.1.1. The optimal trained value

corresponds to the best value that approximates the data. Therefore, to make the model smoother, limiting its ability to model fast changes, the length scale must be set to a larger value.

The first option is to pre-set the length scale to a particular value and prohibit it from optimizing. This approach was successful, but inflexible, as the length scale was the same for all models, and probably not optimal for most of them regarding the task. The second option is to limit the length scale to a specified window. The model then optimizes the length scale, but it stays in the provided range. This approach is advantageous since it is possible to devise reasonable values for the range based on the transit durations.

As for the kernel itself, two variations were used in the experiments: a single SE kernel, which was forced to smooth over the fast changes, and two SE kernels, one modeling the fast changes, while the other modeling the slow changes.

To start with a reasonable length scale, the transit durations for exoplanets from the studied stars were examined. Transit durations vary across different exoplanets, as is shown in a histogram in Figure 4.7. The 99% percentile from this data is equal to approximately 11 hours, which was used as a starting point for experiments with different length scales.

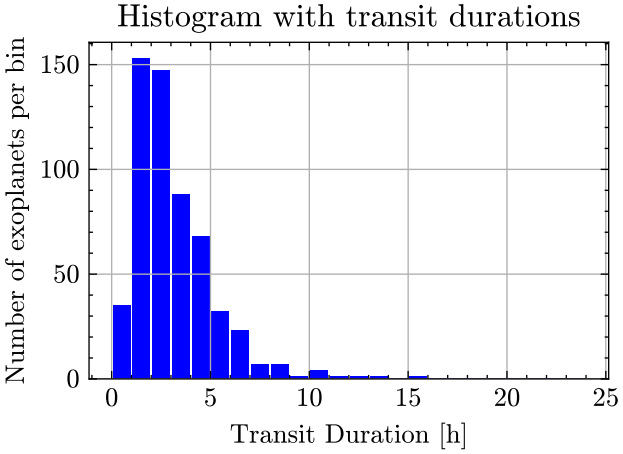

Figure 4.7: Transit durations for confirmed exoplanets from the analysed data.

Since the goal is not to model the fast trends in the data, downsampled data was used, averaging every 3 and later 5 data points. This also helped to shorten the training time for the models. The used length scale ranged from a couple of minutes to a couple of hours for the fast kernel, and half a day to 2 days for the slow kernel. The single kernel was created with the same parameters as the slow kernel.

A common problem for both approaches, more prominent in the single-kernel approach, arose when the model even slightly modeled the transit, creating a slight dip around the transit. This results in a spike around the transit in the detrended light curve, which also threw off the categorizing process because the difference between the original and detrended light curve was now higher than it would be without the transits. It was therefore necessary to minimize the occurrence/magnitude of this happening. The composite kernel showed better results for this problem because, as the fast kernel modeled the transits, the slow kernel could focus on the slower trend, and the transits were already accounted for in the model. The single kernel models, however, still modeled the transits to some extent because

they tried to fit the data best, including the transits. Figure 4.8 shows examples of the model capturing the transits and the final detrended light curve.

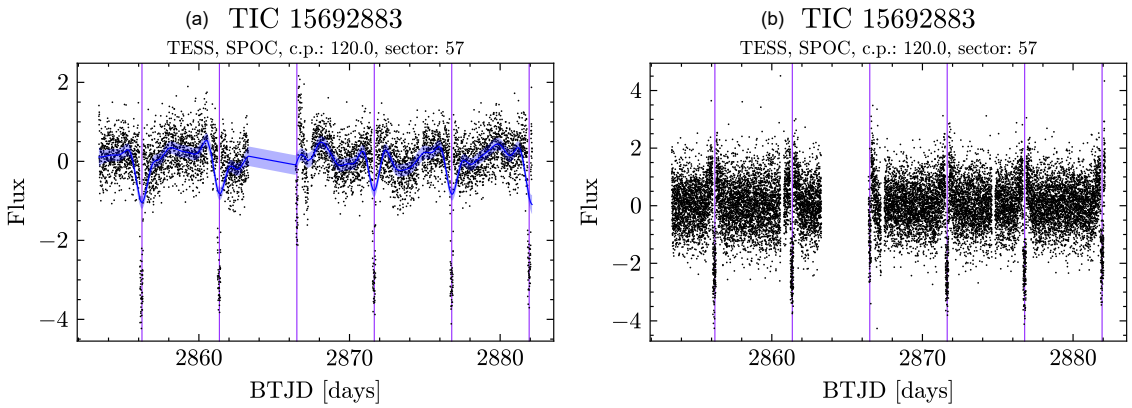

Figure 4.8: Example of (a) a light curve and its detrending model and (b) the detrended light curve. The model creates dips around the transits, which produce peaks in the detrended light curve.

In general, setting the lengthscale to a larger value can result in completely flat models. This can be beneficial, as it can be better to leave the light curve as is if the original data is flat enough or contains only a small amount of stellar variability, than to remove any trend (and possibly the transits). However, when the length scale is too high, the stellar variability may not get modeled properly, or at all, in light curves that do need detrending.

Models with the composite kernel ignored the transits better, as they were captured by the fast kernel, and the slow kernel primarily focused on the stellar variability. However, in certain cases, more rapid changes in the stellar variability were problematic, as the slow kernel did not necessarily learn them. Figure 4.9 shows the kernel decomposed into the fast and slow SE kernels.

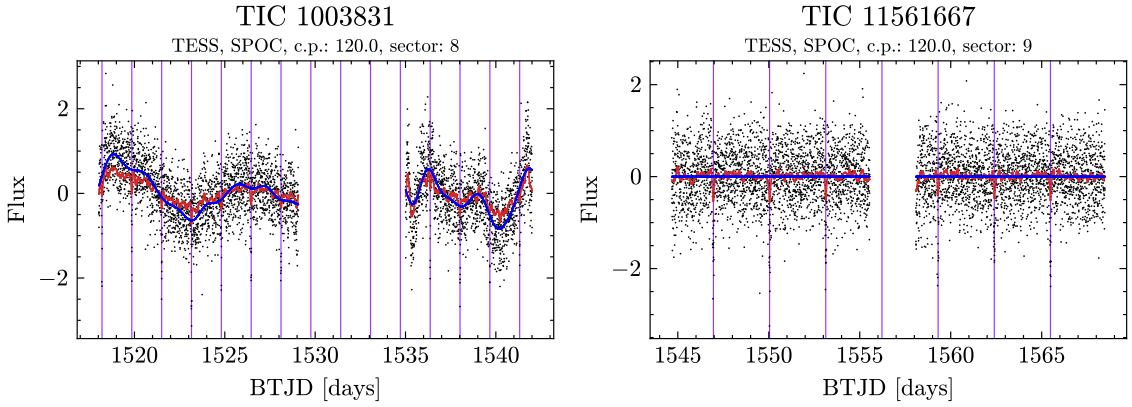

Figure 4.9: Two examples of the detrending model, decomposed into the kernels. The red line is the fast SE kernel, with lower length scale, while the blue is the slow SE kernel, with higher length scale.

Figure 4.10 shows examples of detrended light curves using the composite kernel, with length scale corresponding to 1 to 5 hours for the fast kernel, and 0.5 to 1.5 days for the slow kernel.

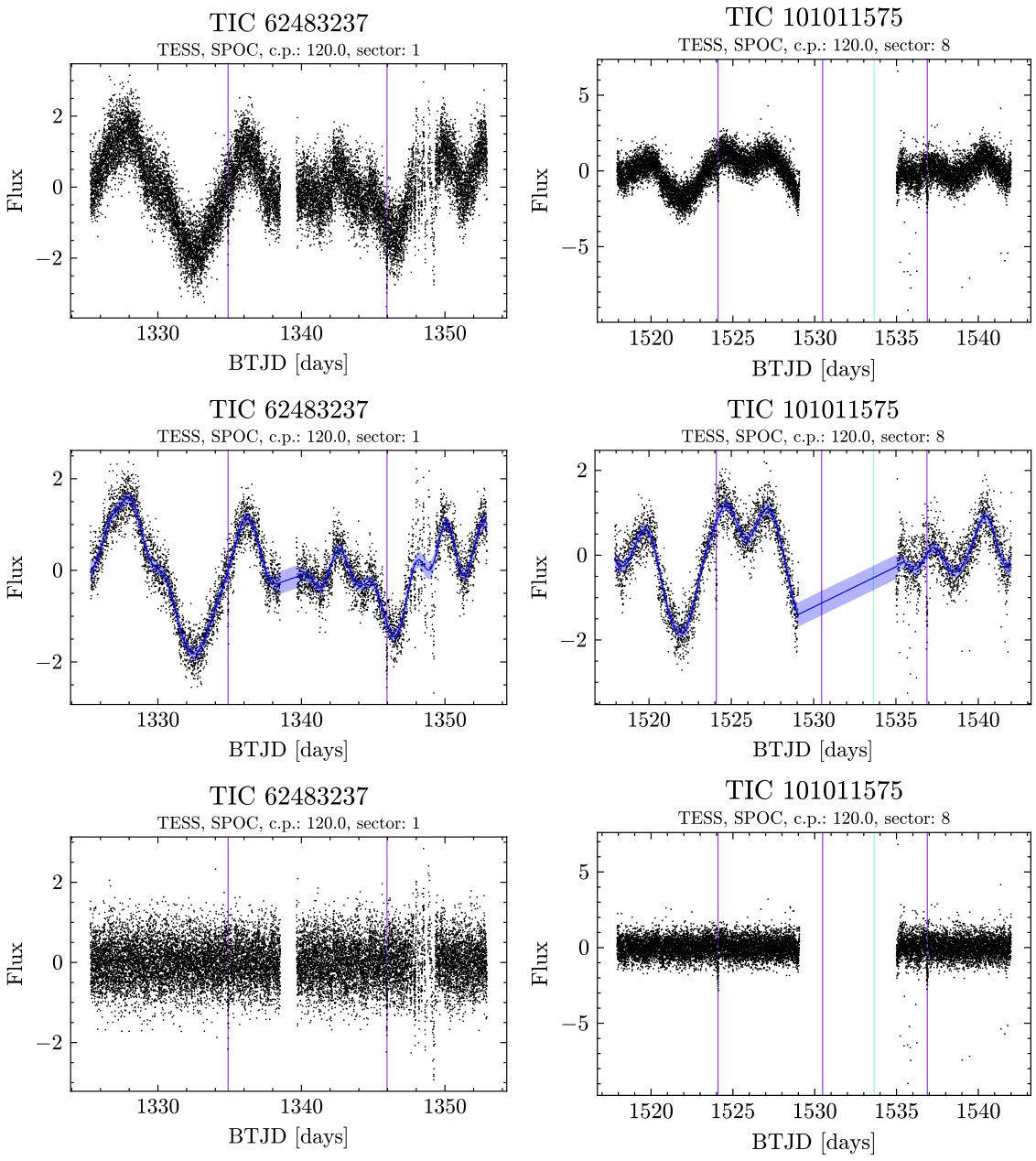

Figure 4.10: Examples of detrending on two different light curves. The first row shows the original light curves, while the second shows the detrending models, and the third the final detrended light curves.

4.2 Detection of transits

This section describes the three examined approaches for the detection of transits in a light curve.

4.2.1 Fourier transform

The first option was to analyze the frequency domain of the light curve. However, a light curve is an unevenly spaced time series, which poses a non-trivial problem with the correct calculation and interpretation of the frequency spectrum [35], and is outside of the scope of this work. The missing values were filled with the mean value (which is zero since the data is normalized). Then, the frequency with the highest peak in the spectrum was selected and interpreted as the found period. Results of this are shown in Figure 4.11. It is clear that this approach is not precise enough to get any meaningful results for all but a few cases.

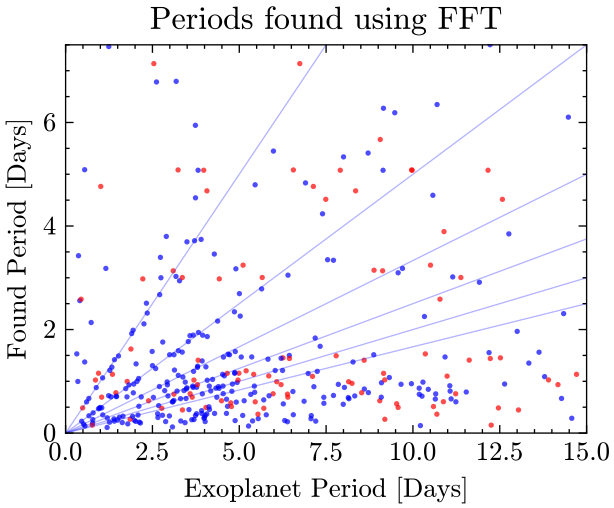

Figure 4.11: Real periods and found periods for exoplanets. Lines show periods corresponding to harmonic frequencies. Red exoplanets come from a multi-exoplanet system – transits from multiple stars can be present in a single light curve.

4.2.2 GPR with periodic kernel

The periodic kernel looks for periodicity in the data, which can be useful when detecting exoplanets with short periods. However, in multi-exoplanet systems, where the individual transits are not equally spaced in time, or with exoplanets with long periods, the periodic kernel reveals little information. In fact, the periodic kernel pushes the periodicity to the data, even when there is none relevant to the detection, as is shown in Figure 4.12. This results in models overproducing transits, which do not match the original data.

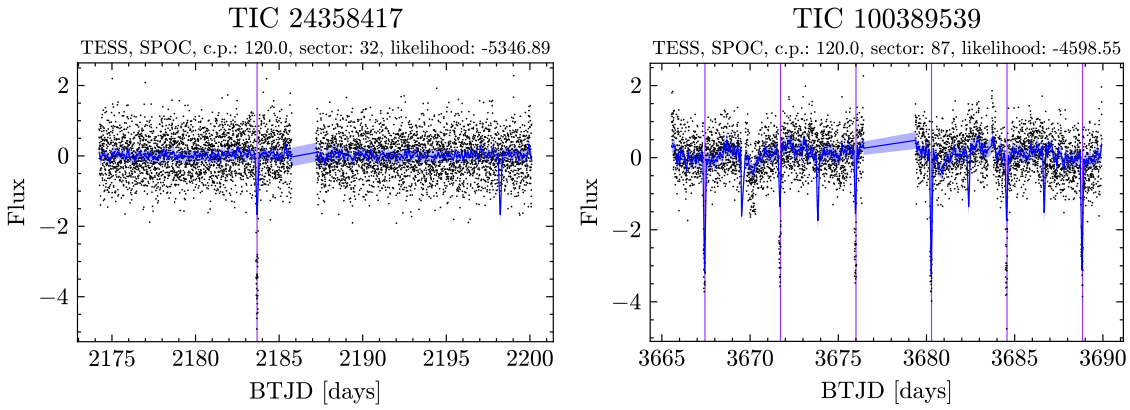

Figure 4.12: Two examples of light curves modeled with the periodic kernel.

4.2.3 Correlation with folded transit-based models

For each star, information about the detected exoplanet is available as well. The key attributes are the transit period, duration, and midpoint. The transit midpoint is a point in time marking the middle of a transit. With this information, a light curve can be folded over on itself, producing a stacked view over the transit. Examples of the folded transits are in Figure 4.13.

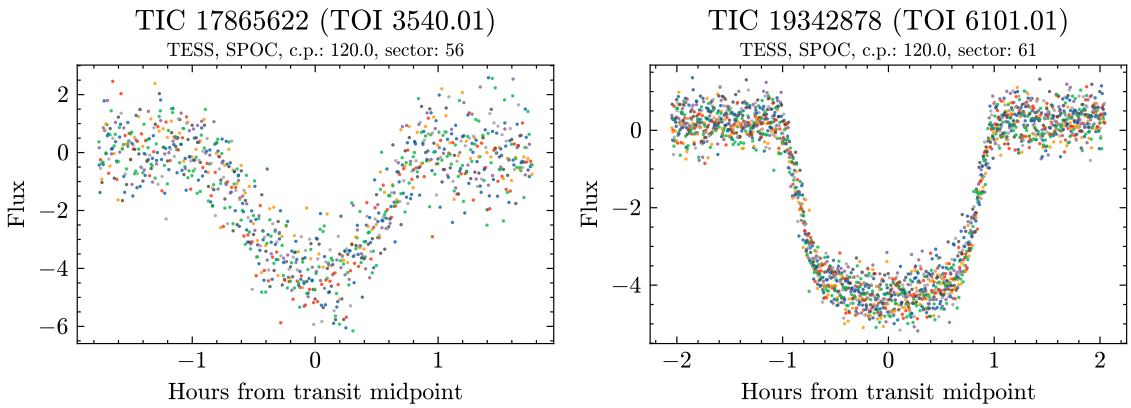

Figure 4.13: Folded light curves on transits for two different stars. Colors differentiate between each instance of a transit.

The next step was to create a model from the transit data. The SE kernel was used, as there is no particular requirement on the model aside from smoothness. Examples of models for folded light curves from Figure 4.13 are shown in Figure 4.14.

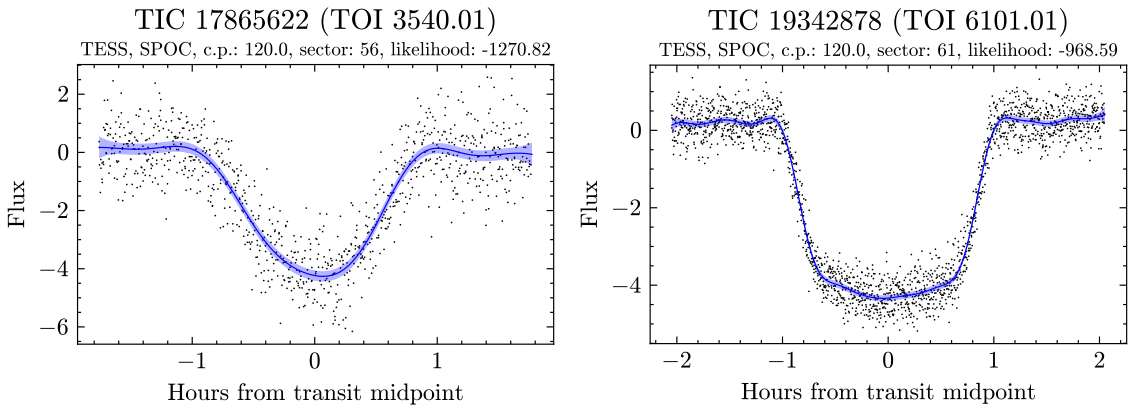

Figure 4.14: Models of folded light curves for two different stars.

Predictions from these models can then be used to calculate the correlation between a light curve and the transit. If the light curve contains transit-shaped events, the correlation will produce higher values than in other parts. The final step was to create a threshold for the correlation values. In places where the correlation exceeded the threshold, a transit was marked as detected. Figure 4.15 shows the individual steps from this approach.

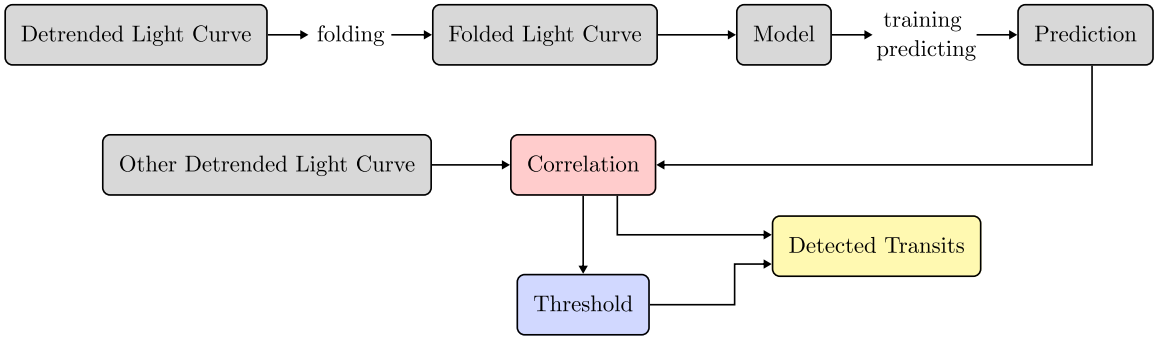

Figure 4.15: The detection process using correlation.

The benefit of this approach is that, unlike with FFT or the periodic kernel, there is no requirement for periodicity in the input data. This can be useful for the detection of exoplanets with extensive orbit periods, with only a few transits available in the total observations, or in systems with multiple orbiting systems. On top of that, each known exoplanet can produce its transit model. A light curve can then be correlated with many different transit models, which can account for various parameters of the star-exoplanet system, as they greatly influence the transit shape.

For the detection of the actual peaks in the correlation, a moving average was used, with a gap around the inspected point, so that the threshold at that point is not contaminated with the correlation peak value. This approach is also used by the Constant False Alarm Rate CFAR algorithm, typically used in radars [36]. There are 3 parameters for the threshold: the number of gap cells and reference cells, and bias.

The number of gap cells specifies the number of samples to ignore to the left and right of the target data point. This ensures that data from the peak, which is to be detected, is not included in the calculation of the threshold. The reference cells are the data points used for the calculation of the threshold and lie beyond the gap cells. These data points

are then averaged, producing the final threshold. A large number of reference cells ensures a smooth threshold, while a small number creates a threshold that is very reactive to the surroundings of each data point. The last parameter is bias and is used as a multiplier of the threshold to move it along the y-axis.

Figure 4.16 shows examples of the correlation and thresholding. The correlation is z-score normalized, which makes it easy to overlay over the data. Values below 1 were set to 1, which significantly reduces the noise, making the thresholding more efficient. The benefit of using the adaptive thresholding approach instead of just a static threshold is that in places of high noise in the correlation, the threshold is increased as well, as there are many data points near each other with higher values.

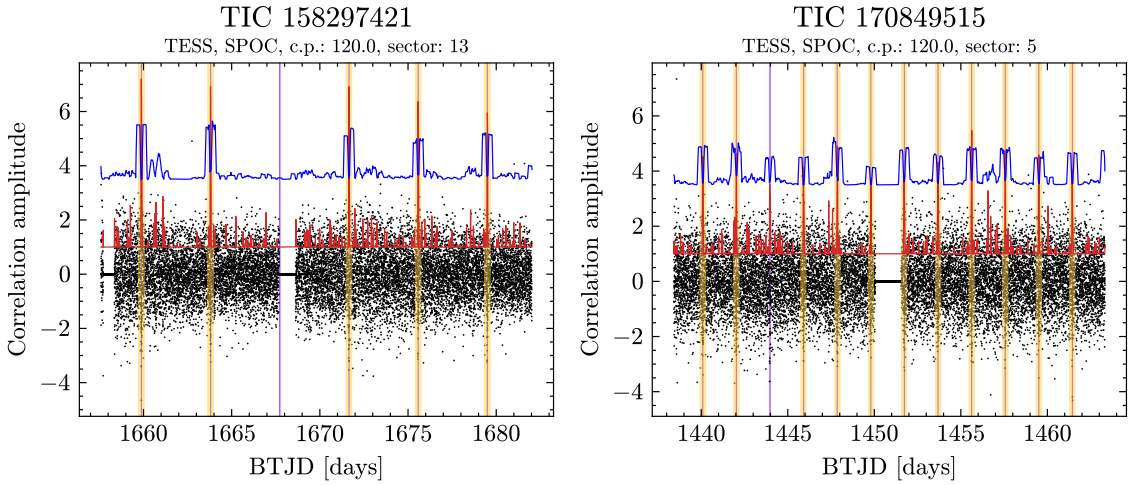

Figure 4.16: Examples of correlation and detection results over two different light curves. The red line shows the correlation, while the blue line shows the threshold. Yellow lines mark detected transits.

To assess the effectiveness of this approach, the transit model was evaluated against every light curve (excluding the one from which the transit model was created). Then, various scores from equations 4.1, 4.2, and 4.3 were calculated for each model, considering the model's ability to detect a transit, and also taking into account if the transit is correct – if there is a real transit in close proximity. Parameters for the threshold were the same as for the final pipeline. The best-performing model was from exoplanet TOI 2193.01, with precision 81.34%, recall 34.02%, and F1 score 47.97%, and is shown in Figure 4.17. The calculations were made using the following formula:

$$\text{precision} = \frac{\text{correctly detected}}{\text{detected}} \tag{4.1}$$

$$\text{recall} = \frac{\text{correctly detected}}{\text{detectable}} \tag{4.2}$$

$$\text{F1 score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{4.3}$$

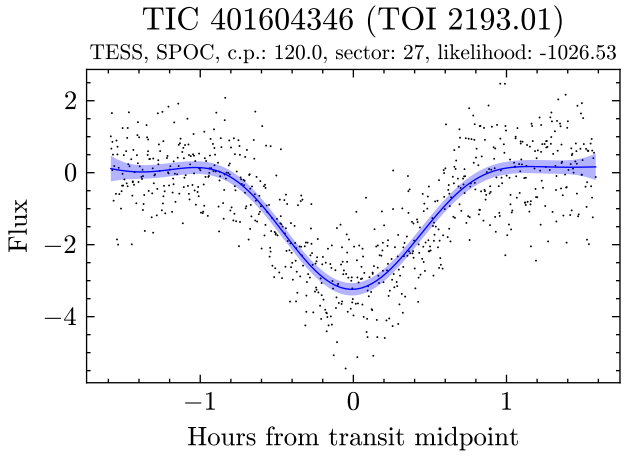

Figure 4.17: Best-performing transit model.

4.3 The final pipeline

The last step was to build a pipeline, evaluating thousands of light curves, and producing statistical results. This section describes the data used, the pipeline, and the results.

4.3.1 The input data

For the final analysis, data from TESS SPOC [5] were chosen, with the list of observed stars for each sector available at `https://archive.stsci.edu/hlsp/tess-spoc`. Many stars are observed as part of multiple sectors, as described in Section 2.5.2. The stars were sorted by the number of observations; all light curves from the stars with the most observations were used. Therefore, for a single star, multiple light curves were processed and, in the analysis of the results, combined to look for exoplanets with long orbital periods. Figure 4.18 shows a histogram of observations per star from SPOC. In total, over 9000 unique stars were analyzed, with a total of over 110000 light curves.

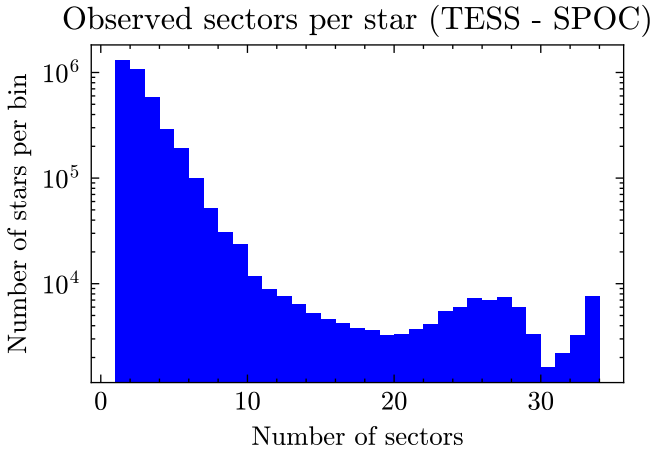

Figure 4.18: Observations counts per star from SPOC.

Transit models

Transit models were constructed from the 405 stars for confirmed exoplanets. The light curves were detrended first, using the same approach as discussed in the following section, but with a lower downsampling rate (averaging every three data points). The detrended light curves were then folded, resulting in 450 transits. Models using the SE kernel were created and trained. Finally, predictions were made with these models over equally spaced timestamps. These predictions correspond to those shown in Figure 4.14, and will be further called transit models.

4.3.2 The data processing pipeline

The data processing pipeline consisted of multiple steps, which can be categorized into several groups:

1. Creating stars – this step includes the creation of the list of stars, producing the Star objects, and downloading the light curves.

2. Creating light curves – with the light curves downloaded, they were put into separate files, as the original versions were needed for detrending further in the pipeline. At the same time, a downsampled version was produced, averaging every 5 cells, which was used for the detrending model.

3. Creating and training models – next, a model using a composite kernel made of a SE kernel with length scale range of 1 to 5 hours, with the initial value of 2 hours, and another SE kernel, with the length scale range of 0.5 to 1.5 days, with the initial value of 1 day, was created with the downsampled data. This model was then trained. After training, the slower kernel was extracted and put into a new model, called the detrending model, which was used to create a prediction over the time values from the original light curve.

4. Detrending – the prediction of the detrending model was subtracted from the original flux values for each light curve. If the mean absolute difference between the original and detrended light curve surpassed 0.1, the detrended light curve was used for detection. Otherwise, the original light curve was used for detection.

5. Correlation – the next step was computing the correlation with the transit models. The 405 stars produced 450 transit models in total. The correlation outputted JSON files with found transit positions, quality score for each detected transit, computed as the max correlation divided by the mean threshold (taken from the whole region where the correlation was higher than the threshold). The thresholding values were set to 1.5 times the width of the transit model for the gap cells, and 6 times for the reference cells. The bias value was set to 4.

6. Analysis of the results – the last step included the analysis of the results from the correlation. The details of this step are discussed in the following section.

Figure 4.19 shows the diagram of the pipeline. At the same time, plots from detrending and correlation were produced.

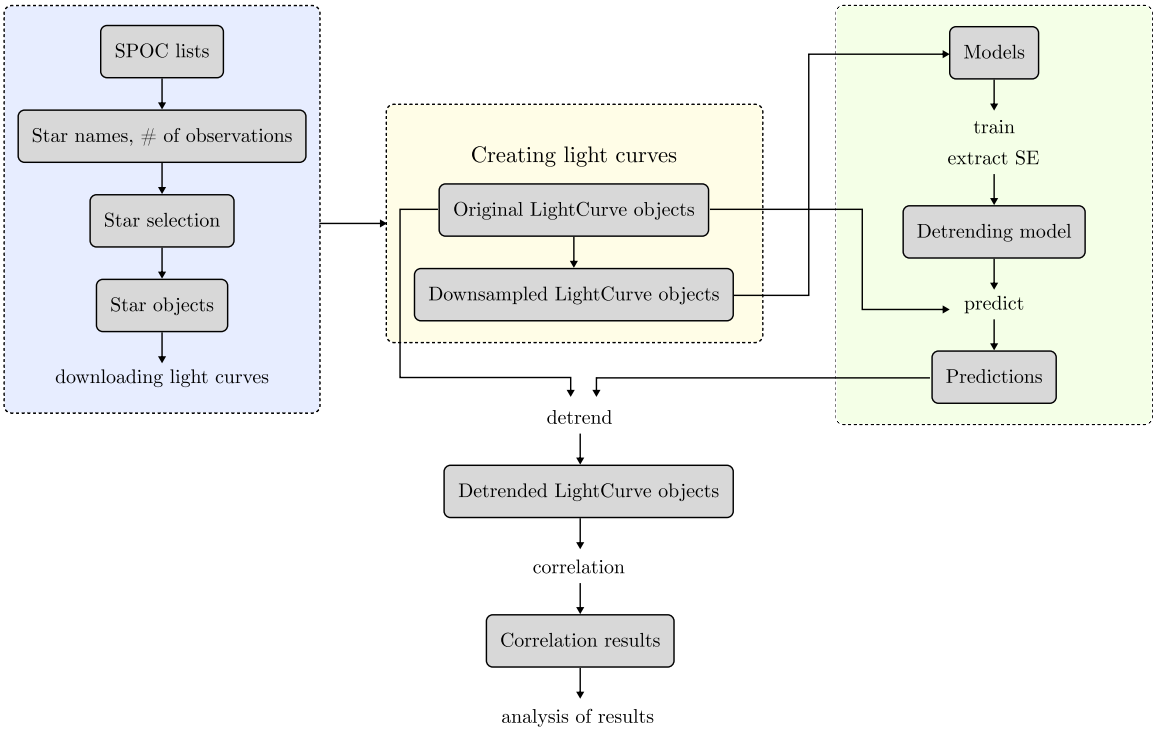

Figure 4.19: The final data processing pipeline.

4.3.3 Results

Figure 4.20 shows the counts of detected transits in the data. Tables 4.1 and 4.2 show results for stars with the highest count of detected transits, and with the highest average correlation to threshold ratio, respectively. It is important to note that the quality of the transit model was not taken into account in these statistics. Intuitively, low-quality models may over-detect possible transits and significantly skew the results.

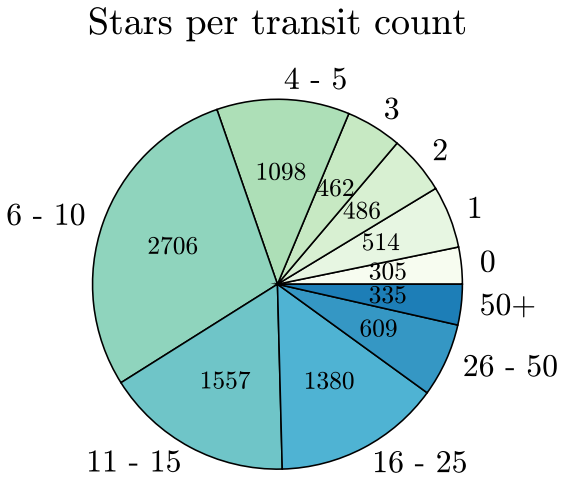

Figure 4.20: Stars per transit count.

Star name	Number of detected transits	Transit model
TIC 229671380	838	TOI 731.01
TIC 420114036	694	TOI 731.01
TIC 33864821	583	TOI 620.01
TIC 329248002	485	TOI 2364.01
TIC 306828113	466	TOI 731.01

Table 4.1: Stars with the highest number of detected transits. All transit models applied.

Star name	Correlation / Threshold	Transit model
TIC 177160238	9.7513	TOI 1696.01
TIC 388203226	9.2889	TOI 1696.01
TIC 287138263	7.1965	TOI 1696.01
TIC 288510643	7.1769	TOI 136.01
TIC 165551136	6.9278	TOI 1696.01

Table 4.2: Stars with the highest average correlation to threshold ratio. All transit models applied.

As stated before, these raw results can be misleading, since there was no prior evaluation of the transit model. In fact, the transit models from Table 4.1 and 4.2, due to their too generic shape, ensure a high rate of false positive detections. To obtain actionable results, the relevant transit models were examined and removed from the dataset if deemed low-quality. Tables 4.3 and 4.4 show similar results, but only with transit models examined.

Star name	Number of detected transits	Transit model
TIC 33864821	449	TOI 564.01
TIC 329248002	434	TOI 1454.01
TIC 236761861	370	TOI 129.01
TIC 306828113	347	TOI 129.01
TIC 150437346	333	TOI 564.01

Table 4.3: Stars with the highest number of detected transits. Transit models verified.

Star name	Correlation / Threshold	Transit model
TIC 284196801	4.3807	TOI 1073.01
TIC 388203195	4.2696	TOI 129.01
TIC 141094672	3.7159	TOI 564.01
TIC 33834564	3.559	TOI 674.01
TIC 177308817	3.5458	TOI 5398.01

Table 4.4: Stars with the highest average correlation to threshold ratio. Transit models verified.

By analyzing data from all the available sectors for a star, it is possible to look for long-periodic exoplanets. To analyze the transit period, we are looking for a minimum of four transits to be detected in the light curves across all the used sectors. Table 4.5 shows stars with the longest average period found.

Star name	Period [day]	Transit model	Data range [day]
TIC 453101762	921.0037	TOI 123.01	2421.6498
TIC 349909614	916.0423	TOI 778.01	2421.6501
TIC 350144739	915.4716	TOI 1836.01	2421.6499
TIC 300159098	914.211	TOI 163.01	2421.6489
TIC 220397755	910.3146	TOI 1516.01	2361.006

Table 4.5: Stars with the highest average period. Transit models verified.

4.3.4 Performance

The level of down-sampling had a great impact on the time spent on training and memory usage, as is shown in Figure 4.21. Therefore, models for detrending were trained on down-sampled light curves, using gap-aware moving average with various window sizes.

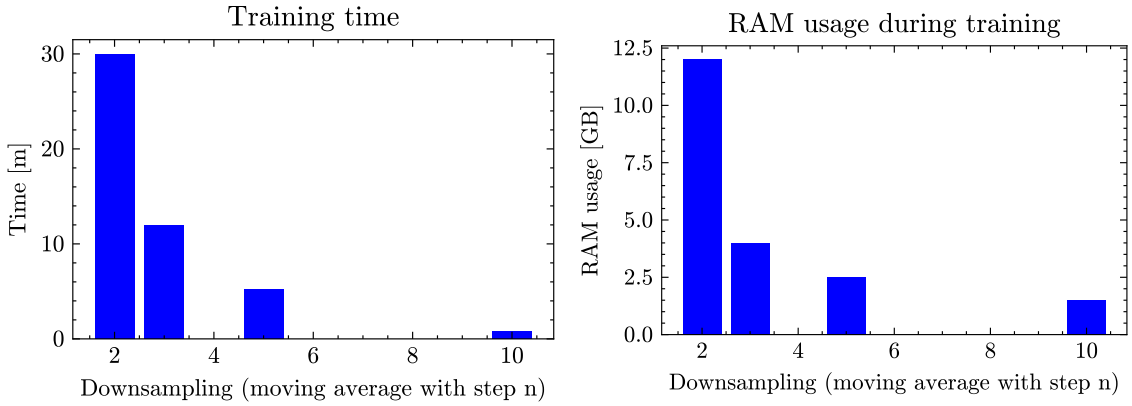

Figure 4.21: The training time and memory usage grow significantly with the number of training samples. This is expected due to the GPR complexity mentioned in 3.1.

The correlation was also non-trivial in regards to the computational time due to the number of transit models used. For a single light curve, correlation with 450 transit models took approximately 5 minutes on average.

Chapter 5

Implementation

The implementation of the whole data processing pipeline is written in Python. The main part is a Python package called `exo_search`, consisting of multiple sub-packages. The other part is a command-line interface, allowing to build pipelines.

`exo_search` contains the following sub-packages:

- `exo_search.entities` – classes `Star`, `LightCurve`, and `Exoplanet`.
- `exo_search.modeling` – classes `Model`, `Manager`, and `Prediction`.
- `exo_search.pipeline` – scripts for command line interface.
- `exo_search.utils` – general utility functions.

Commands are accessed via script `run.py`. The commands are defined in `cmd_interface/` in `pipeline.py`, `star_lists.py`, `tess.py`, `kepler.py`.

5.1 Main entities classes

Classes `Star`, `LightCurve`, and `Exoplanet`, defined in `exo_search.entities`, are the main data classes. They hold information used for training and evaluation. The main purpose of `Star` is to download light curves from the MAST archive. `LightCurve` holds the timestamps and flux values, and is passed to models for training. `Exoplanet` can be held both by `Star` or `LightCurve`, and is used for plots and evaluation.

All of these classes are serialized into `JSON` files, using custom `to_dict()` and `from_dict()` methods.

5.1.1 `Star` class

`Star` class, defined in `exo_search.entities.star.py`, represents a single star. The main use case is downloading and holding light curves from MAST. `Star` is used at the beginning of the pipeline.

`Star` object holds names associated with the star, along with its exoplanets and light curves. Light curves managed by this class are raw and unfiltered. They are used to construct light curves that are used for training, which are then stored separately. This class is also responsible for communicating with the MAST archive, using the `lightkurve` library. It also holds information about observations available for the star, ready for download.

This class offers methods for getting/filtering light curves, and mainly, downloading. The `download_lc()` method facilitates the whole downloading process. The arguments for this method are the mission, author, and cadence period. It also supports downloading multiple light curves at once. The download process includes obtaining the list of available observations, which is done using the `Star.get_search_result()`. Next, the list is filtered for the provided parameters. The matching observations are compared with already downloaded light curves, using the batch number (sector, quarter). Resulting light curves are then downloaded and converted to `LightCurve` objects. Downloading also supports an iterative approach, since a star can have multiple names, all names from the list of names are checked if the download was unsuccessful. When an error occurs during the download, the method can go to sleep for a defined period of time. A `Star.download_attempts` variable controls how many tries are performed, with `Star.download_sleep`, `Star.download_multiplicator`, and `Star.download_range` control how much time to wait before the next try.

5.1.2 `LightCurve` class

`LightCurve` class, defined in `exo_search.entities.light_curve.py`, holds data and metadata for light curves. Light curves can be combined, downsampled, filtered, and folded for examination and training. `LightCurve` objects also hold the metadata – mission, author, cadence period. If multiple light curves are combined, the resulting `LightCurve` object holds information about the combined batches.

A list of selected methods available in the `LightCurve` class and their description follows:

- `plot()` – Plots the time and flux values as a scatter plot, optionally marking transit positions. Exoplanets are plotted using `_plot_exoplanets()`.
- `_plot_exoplanets()` – Add vertical lines corresponding to transit positions to an existing axis. Transit positions are calculated using `get_transits()`. Individual exoplanets are differentiated by color. If the exoplanet is not confirmed, the transit lines are dashed. This method accepts the axis as an argument, so that it can be used in plots of models as well.
- `join_lc()` – Joins multiple `LightCurve` objects into one. The flux is automatically normalized, as the absolute values can differ across batches.
- `get_transits()` – Calculates transit positions in the current light curve for provided `Exoplanet` object.
- `fill_gaps()` – Fills gaps in time and flux with `np.nan`. The gap is calculated from the cadence period, but can be provided as a parameter. This method performs best on the original data, which contains gaps due to bad cadences being filtered out.

5.1.3 `Exoplanet` class

`Exoplanet` class, defined in `exo_search.entities.exoplanet.py`, holds metadata about exoplanets, including a list of names, orbit period, disposition (e.g. confirmed, false positive), transit midpoint, and transit duration. Exoplanets are stored with the corresponding `Star` object, and copied to the `LightCurve` object if it is being stored separately.

5.2 Modeling related classes

To manage large quantities of models, the class `Manager` was implemented, which makes writing custom scripts working with models in various ways convenient. `Manager` provides methods for triggering specific functions for the whole collection of models.

Predictions made by the models are stored in the form of `Prediction` objects. A model can produce more than one distinct prediction. Each one is represented by `Prediction` object, which also manages the serialization of the data.

5.2.1 `Model` class

`Model` class, defined in `exo_search.modeling.model.py`, is a class representing a single model. The class offers methods for serialization/deserialization, model creation, training, predicting, and plotting. This class also holds the `LightCurve` object used for training and selected model parameters. The actual GPR model is created using `gpflow` package (further discussed in Section 5.5). A `Model` object also holds a list of its predictions.

The GPR `gpflow` model is stored as pickle using the `cloudpickle` library, because although `gpflow` offers ways to serialize the models, the type of kernel, and any complex priors are not being serialized using any of the recommended ways. This means that each time a model would be loaded, new kernel of the exact same type and with all the priors would need to be created, for the model to load its parameters into it. This would result in an inflexible solution, not allowing for models with different kernels to reuse the same code for training, plotting, detrending, etc.

Follows a list of selected methods available in the `Model` class and their description:

- `train()` – the `gpflow` model is trained. After training, the likelihood, its variance, and the found period (if the kernel is periodic) are set as attributes of the `Model` object. The training is done by minimizing the training loss of `gpflow.optimizers.Scipy` optimizer.
- `predict()` – checks for already existing prediction with the same x values. If none match, a new one is created. By default, only the f-values are predicted. Optionally, y-values are predicted as well.
- `plot()` – plots a provided prediction and its 95% confidence interval for f-values and y-values.

5.2.2 `Manager` class

`Manager` class, defined in `exo_search.modeling.manager.py`, is responsible for managing a group of models. It provides methods for manipulation with all models, as well as filtering, plotting, loading, and storing.

A list of selected methods available in the `Manager` class and their description follows:

- `create_models_from_lc()` – creates new `Model` objects from provided light curves.
- `color_models_by_star()` – based on labels created using the `LabelMe` [43] software and provided mapping from label to color, assigns the color and the label to each model. This color and label can then be used in `plot_model_properties()`.

- `plot_model_attributes()` – plots the properties of the models against each other. Three types of attributes are available: model attributes, prediction attributes, and kernel attributes. All of these have pre-set values, but can be customized by providing custom lists for each one. A list of operations to perform over the prediction attributes can be provided as an attribute as well.

5.2.3 `Prediction` class

`Prediction` class, defined in `exo_search.modeling.prediction.py`, represents a single prediction from a model. Each model contains a list of its predictions, which can be stored and loaded. Predictions are then used for plots and subsequent analysis of the model.

5.3 Pipeline commands

The commands for the pipeline are defined in `exo_search.pipeline`. The sub-package contains scripts for the most common operations with models, creating stars, light curves, predictions, generating plots, etc. Each script has a `main()` function and optionally a set of supporting functions.

The commands are accessible via `run.py`, or any script in `cmd_pipeline/`, depending on the category.

A list of selected commands/command categories follows:

- `create_*` – creating stars, light curves, and models.
- `plot_*` – plotting light curves, models, models' attributes, detrending, etc.
- `predict`, `train` – manipulating models.
- `fold` – each light curve is folded for each of its exoplanets around its transits. The transits are located using the `LightCurve.get_transits()` method. Parts of the light curve centered around transits with size of two transit widths are stacked onto themselves. The folded light curve is saved as a new `LightCurve` object. The command also allow the creation of plots showing the transit with colors differentiating each instance.
- `correlate` – detecting transits using the correlation method. Produces plots showing the calculated correlation, threshold, and found transits, and a `JSON` file with the results.

Besides the main plots and objects, some of the commands produce a list of results for each parsed entity. For example, the `detrend` command produces a list containing a boolean value for each light curve, marking if the light curve was detrended based on the provided criteria, or not. The `correlate` command produces a list of time stamps of detected transits for each light curve for each model. The highest correlation value divided by the mean value of the threshold in the region where correlation is higher than the threshold is saved for each transit position as well. This value is also saved in cases when there is no transit detected in the light curve, but only the highest value; the timestamp for this point is saved as well. This data can then be used to assess the found periods in the transits, the confidence in those points, etc. For data with known exoplanets, counts of how many transits were correctly detected can be produced as well.

5.4 Logging

Some information is printed to the standard output for convenience. More detailed information is outputted using the `Logger` class, defined in `exo_search.utils.logger.py`. This class utilizes the library `filelock`, preventing multiple instances of the code from writing into the log file at the same time, potentially corrupting the logs.

5.5 External libraries

The code was developed in a virtual environment `venv` with `Python` 3.10. `requirements.txt` contains libraries and their versions from this environment. Some notable libraries include:

- `lightkurve` [23] (v. 2.5.0) – searching for and downloading light curves. This package builds upon multiple packages widely used in data processing in astronomy and astrophysics [1] [2] [3] [4] [13].
- `gpflow` [24] (v. 2.9.2) – creating, training, and making predictions with the GPR model.
- `scienceplots` [12] (v. 2.1.1) – styling `matplotlib` plots for better readability on paper.
- `cloudpickle`[1] (v. 3.1.1) – pickling the `gpflow kernel` and the model parameters.
- `typer` [32] (v. 0.15.2) – managing command line arguments.
- `filelock`[2] (v. 3.18.0) – locking the log file during writes.

Other heavily used libraries include `numpy` [16] (v. 2.1.3), `pandas` [25] [37] (v. 2.2.3), and `matplotlib` [18] (v. 3.10.1).

[1] https://github.com/cloudpipe/cloudpickle
[2] https://py-filelock.readthedocs.io/en/latest/index.html

Chapter 6

Conclusion

The goal of this work was to create an automated way of analyzing light curves and detecting transits. This will help to identify promising stars for further observations.

In the first step of this thesis, the commonly used exoplanet detection methods were examined. For further analysis, the transit method was chosen due to the large quantities of available data, specifics of which are also described at the beginning of this work.

The chosen approach was surrogate modeling, specifically GPR. Using GPR, various models of the data were constructed and trained. These models were then used for multiple tasks during the analysis.

The next step includes analysis of the data. There were three components of this part of the work – categorization of light curves based on the presence of stellar variability, detrending, and detection of transits. For each step, multiple approaches were assessed. Finally, a pipeline from the devised steps was constructed and used to analyze over a hundred thousand light curves, searching for long-periodic exoplanets. Results from this analysis were passed to the Astronomical Institute of CAS for further observations using the radial velocity method.

The final implementation includes a flexible Python library that allows for manipulating light curves and creating models. It also offers a set of commands suitable for making data processing pipelines.

Future work could implement support for analysis of radial velocity measurements and tools for a combined assessment of results for both exoplanet detection methods. The quality of the transit model could be evaluated automatically as well. Lastly, using information about the found periods in the light curves, multi-exoplanet systems could be examined and detected.

During the work on this thesis, I improved my Python knowledge, learned about surrogate modeling and GPR, and explored working with large quantities of data. I am looking forward to future cooperation with the Astronomical Institute of CAS to further improve my thesis results.

Bibliography

[1] Astropy Collaboration; Price-Whelan, A. M.; Sipőcz, B. M.; Günther, H. M.; Lim, P. L. et al. The Astropy Project: Building an Open-science Project and Status of the v2.0 Core Package. *The Astronomical Journal*, september 2018, vol. 156, no. 3, p. 123. Available at: https://doi.org/10.3847/1538-3881/aabc4f.

[2] Astropy Collaboration; Price-Whelan, A. M.; Lim, P. L.; Earl, N.; Starkman, N. et al. The Astropy Project: Sustaining and Growing a Community-oriented Open-source Project and the Latest Major Release (v5.0) of the Core Package. *The Astrophysical Journal*, august 2022, vol. 935, no. 2, p. 167. Available at: https://doi.org/10.3847/1538-4357/ac7c74.

[3] Astropy Collaboration; Robitaille, T. P.; Tollerud, E. J.; Greenfield, P.; Droettboom, M. et al. Astropy: A community Python package for astronomy. *Astronomy & Astrophysics*, october 2013, vol. 558, p. A33. Available at: https://dx.doi.org/10.1051/0004-6361/201322068.

[4] Brasseur, C. E.; Phillip, C.; Fleming, S. W.; Mullally, S. E. and White, R. L. *Astrocut: Tools for creating cutouts of TESS images* Astrophysics Source Code Library, record ascl:1905.007. May 2019. Available at: https://ui.adsabs.harvard.edu/abs/2019ascl.soft05007B.

[5] Caldwell, D. A.; Tenenbaum, P.; Twicken, J. D.; Jenkins, J. M.; Ting, E. et al. TESS Science Processing Operations Center FFI Target List Products. *Research Notes of the American Astronomical Society*, november 2020, vol. 4, no. 11, p. 201. Available at: https://doi.org/10.3847/2515-5172/abc9b3.

[6] Cameron, A. C. Extrasolar Planetary Transits. In: Bozza, V.; Mancini, L. and Sozzetti, A., ed. *Methods of Detecting Exoplanets: 1st Advanced School on Exoplanetary Science*. Cham: Springer International Publishing, 2016, p. 89–131. ISBN 978-3-319-27458-4. Available at: https://doi.org/10.1007/978-3-319-27458-4_2.

[7] ESO. *ESPRESSO: Echelle SPectrograph for Rocky Exoplanet and Stable Spectroscopic Observations* https://www.eso.org/public/teles-instr/paranal-observatory/vlt/vlt-instr/espresso/. [cit. 2025-05-10].

[8] ESO. *Yes, it is the Image of an Exoplanet: Astronomers Confirm the First Image of a Planet Outside of Our Solar System* https://www.eso.org/public/news/eso0515/. 30. april 2005. [cit. 2025-05-10].

[9] exoplanets.org. *Exoplanets Data Explorer: 18 Del b* http://exoplanets.org/detail/18_Del_b. [cit. 2025-05-10].

[10] Forrester, A.; Sobester, A. and Keane, A. Constructing a Surrogate. In: *Engineering Design via Surrogate Modelling*. John Wiley & Sons, Ltd, 2008, chap. 2, p. 33–76. ISBN 9780470770801. Available at: https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470770801.ch2.

[11] Forrester, A.; Sobester, A. and Keane, A. Front Matter. In: *Engineering Design via Surrogate Modelling*. John Wiley & Sons, Ltd, 2008, p. i–xviii. ISBN 9780470770801. Available at: https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470770801.fmatter.

[12] Garrett, J. D. garrettj403/SciencePlots. Zenodo, sep 2021. Available at: http://doi.org/10.5281/zenodo.4106649.

[13] Ginsburg, A.; Sipőcz, B. M.; Brasseur, C. E.; Cowperthwaite, P. S.; Craig, M. W. et al. astroquery: An Astronomical Web-querying Package in Python. *The Astronomical Journal*, march 2019, vol. 157, p. 98. Available at: https://doi.org/10.3847/1538-3881/aafc33.

[14] Gould, A. Microlensing Planets. In: Bozza, V.; Mancini, L. and Sozzetti, A., ed. *Methods of Detecting Exoplanets: 1st Advanced School on Exoplanetary Science*. Cham: Springer International Publishing, 2016, p. 135–179. ISBN 978-3-319-27458-4. Available at: https://doi.org/10.1007/978-3-319-27458-4_3.

[15] Guerrero, N. M.; Seager, S.; Huang, C. X.; Vanderburg, A.; Soto, A. G. et al. The TESS Objects of Interest Catalog from the TESS Prime Mission. *The Astrophysical Journal Supplement Series*. The American Astronomical Society, jun 2021, vol. 254, no. 2, p. 39. Available at: https://dx.doi.org/10.3847/1538-4365/abefe1.

[16] Harris, C. R.; Millman, K. J.; Walt, S. J. van der; Gommers, R.; Virtanen, P. et al. Array programming with NumPy. *Nature*. Springer Science and Business Media LLC, september 2020, vol. 585, no. 7825, p. 357–362. Available at: https://doi.org/10.1038/s41586-020-2649-2.

[17] Hatzes, A. P. The Radial Velocity Method for the Detection of Exoplanets. In: Bozza, V.; Mancini, L. and Sozzetti, A., ed. *Methods of Detecting Exoplanets: 1st Advanced School on Exoplanetary Science*. Cham: Springer International Publishing, 2016, p. 3–86. ISBN 978-3-319-27458-4. Available at: https://doi.org/10.1007/978-3-319-27458-4_1.

[18] Hunter, J. D. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*. IEEE COMPUTER SOC, 2007, vol. 9, no. 3, p. 90–95. Available at: https://doi.org/10.1109/MCSE.2007.55.

[19] Irwin, P. G. J. Detection Methods and Properties of Known Exoplanets. In: Mason, J. W., ed. *Exoplanets: Detection, Formation, Properties, Habitability*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, p. 1–20. ISBN 978-3-540-74008-7. Available at: https://doi.org/10.1007/978-3-540-74008-7_1.

[20] Jenkins, J. M.; Caldwell, D. A.; Chandrasekaran, H.; Twicken, J. D.; Bryson, S. T. et al. Overview of the Kepler Science Processing Pipeline. *The*

Astrophysical Journal Letters, april 2010, vol. 713, no. 2, p. L87–L91. Available at: https://iopscience.iop.org/article/10.1088/2041-8205/713/2/L87.

[21] Kabáth, P.; Skarka, M.; Sabotta, S.; Guenther, E.; Jones, D. et al. Ondřejov Echelle Spectrograph, Ground Based Support Facility for Exoplanet Missions*. *Publications of the Astronomical Society of the Pacific*. The Astronomical Society of the Pacific, jan 2020, vol. 132, no. 1009, p. 035002. Available at: https://dx.doi.org/10.1088/1538-3873/ab6752.

[22] Koziel, S. and Leifsson, L. Surrogate-Based Modeling and Optimization: Applications in Engineering. In:. Springer New York, 2013. SpringerLink : Bücher. ISBN 9781461475514. Available at: https://books.google.cz/books?id=Df1GAAAAQBAJ.

[23] Lightkurve Collaboration; Cardoso, J. V. d. M.; Hedges, C.; Gully-Santiago, M.; Saunders, N. et al. *Lightkurve: Kepler and TESS time series analysis in Python* Astrophysics Source Code Library. Dec 2018. Available at: http://adsabs.harvard.edu/abs/2018ascl.soft12013L.

[24] Matthews, A. G. d. G.; van der Wilk, M.; Nickson, T.; Fujii, K.; Boukouvalas, A. et al. GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research*, apr 2017, vol. 18, no. 40, p. 1–6. Available at: http://jmlr.org/papers/v18/16-537.html.

[25] McKinney Wes. Data Structures for Statistical Computing in Python. In: Walt Stéfan van der and Millman Jarrod, ed. *Proceedings of the 9th Python in Science Conference*. 2010, p. 56 – 61. Available at: https://doi.org/10.25080/Majora-92bf1922-00a.

[26] Mighell, K. and Cleve, J. V. *K2: Extending Kepler's Power to the Ecliptic: K2 Handbook*. Moffett Field, CA, 94035: NASA Ames Research Center, aug 2020. Available at: https://archive.stsci.edu/files/live/sites/mast/files/home/missions-and-data/k2/_documents/KSCI-19116-003.pdf. KSCI-19116-003.

[27] NASA. *Exoplanet Exploration: Planets Beyond Our Solar System* https://exoplanets.nasa.gov/alien-worlds/historic-timeline/#first-exoplanets-discovered. [cit. 09-05-2025].

[28] NASA Exoplanet Science Institute. *Exoplanet and Candidate Statistics* https://exoplanetarchive.ipac.caltech.edu/docs/counts_detail.html. [cit. 2025-05-10].

[29] NASA Exoplanet Science Institute. *Exoplanet Plots: Confirmed Planets* https://exoplanetarchive.ipac.caltech.edu/exoplanetplots/. [cit. 2025-05-10].

[30] Perryman, M. *The Exoplanet Handbook*. Cambridge University Press, 2018. ISBN 9781108419772. Available at: https://books.google.cz/books?id=ngtmDwAAQBAJ.

[31] Perryman, M. The history of astrometry. *The European Physical Journal H*, 2012, vol. 37, p. 745–792. Available at: https://api.semanticscholar.org/CorpusID:119111979.

[32] Ramírez, S. *Typer* https://github.com/fastapi/typer. Available at: https://typer.tiangolo.com.

[33] Rasmussen, C. E. and Williams, C. K. I. *Gaussian Processes for Machine Learning*. The MIT Press, november 2005. ISBN 9780262256834. Available at: https://doi.org/10.7551/mitpress/3206.001.0001.

[34] Seager, S.; Lissauer, J. J. and Dotson, R. Introduction to Exoplanets. In: *Exoplanets*. University of Arizona Press, 2010, p. 3–14. ISBN 9780816529452. Available at: http://www.jstor.org/stable/j.ctt1814jx6.6.

[35] Selva, J. FFT Interpolation From Nonuniform Samples Lying in a Regular Grid. *IEEE Transactions on Signal Processing*, june 2015, vol. 63, p. 2826. Available at: https://doi.org/10.1109/TSP.2015.2419178.

[36] Shin, J. H. and Choi, Y. Robust Control for the Detection Threshold of CFAR Process in Cluttered Environments. *Sensors*, 2020, vol. 20, no. 14. ISSN 1424-8220. Available at: https://www.mdpi.com/1424-8220/20/14/3904.

[37] The pandas development team. *Pandas-dev/pandas: Pandas*. Zenodo, sep 2024. Available at: https://doi.org/10.5281/zenodo.13819579.

[38] Thompson, S. E.; Fraquelli, D.; Van Cleve, J. E. and Caldwell, D. A. *Kepler Archive Manual* Kepler Science Document KDMC-10008-006, id. 9. Edited by Faith Abney, Dwight Sanderfer, Michael R. Haas, and Steve B. Howell. May 2016. Available at: https://archive.stsci.edu/files/live/sites/mast/files/home/missions-and-data/kepler/_documents/archive_manual.pdf.

[39] Valio, A. The impact of stellar activity on orbiting planets. *Boletim da Sociedade Astronômica Brasileira*, 2020, vol. 32, no. 1, p. 3–9. Available at: https://sab-astro.org.br/wp-content/uploads/2020/04/AdrianaValio.pdf.

[40] Van Cleve, J. E. and Caldwell, D. A. *Kepler Instrument Handbook* Kepler Science Document KSCI-19033-002, id.1. Edited by Michael R. Haas and Steve B. Howell. April 2016. 1 p. Available at: https://archive.stsci.edu/files/live/sites/mast/files/home/missions-and-data/kepler/_documents/KSCI-19033-002-instrument-hb.pdf.

[41] Van Cleve, J. E.; Christiansen, J. L.; Jenkins, J. M.; Caldwell, D. A.; Barclay, T. et al. *Kepler Data Characteristics Handbook* Kepler Science Document KSCI-19040-005, id. 2. Edited by Doug Caldwell, Jon M. Jenkins, Michael R. Haas and Natalie Batalha. December 2016. Available at: https://archive.stsci.edu/files/live/sites/mast/files/home/missions-and-data/kepler/_documents/Data_Characteristics.pdf.

[42] Vanderspek, R.; Doty, J. P.; Fausnaugh, M.; Villaseñor, J. N. S.; Jenkins, J. M. et al. *TESS Instrument Handbook*. Dec 2018. Available at: https://archive.stsci.edu/files/live/sites/mast/files/home/missions-and-data/active-missions/tess/_documents/TESS_Instrument_Handbook_v0.1.pdf. Version: 0.1.

[43] Wada, K. *Labelme: Image Polygonal Annotation with Python*. 18. November 2021. Available at: https://github.com/wkentaro/labelme.

[44] Winn, J. N. and Dotson, R. Exoplanet Transits and Occultations. In: *Exoplanets*. University of Arizona Press, 2010, p. 55–78. ISBN 9780816529452. Available at: http://www.jstor.org/stable/j.ctt1814jx6.9.

[45] Wolszczan, A. and Frail, D. A. A planetary system around the millisecond pulsar PSR1257 + 12. *Nature*, 1992, vol. 355, no. 6356, p. 145–147. Available at: https://doi.org/10.1038/355145a0.